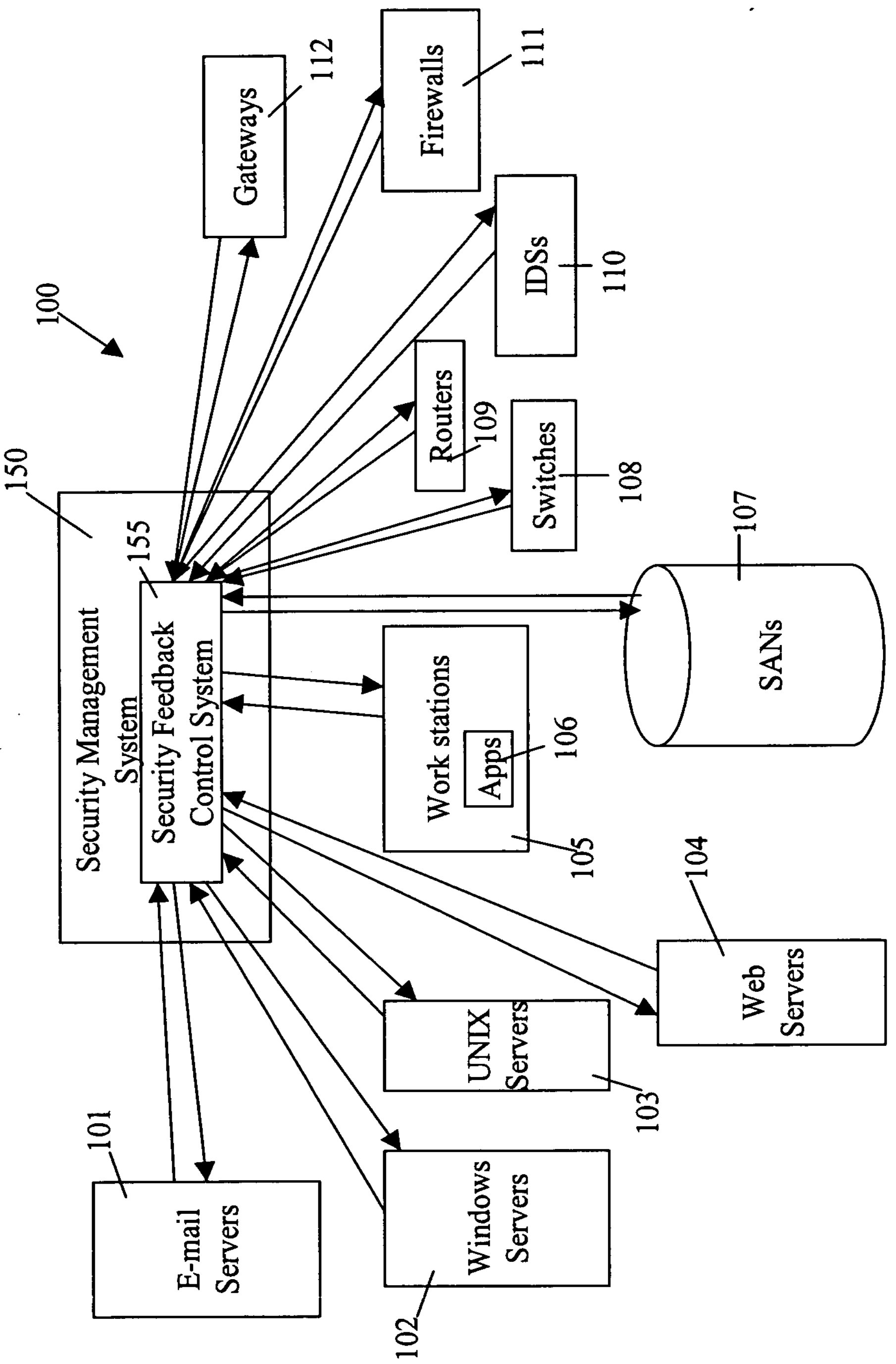

100
150
Security Management System
155
Security Feedback Control System
Gateways
112
Firewalls
111
IDSs
110
Routers
109
Switches
108
SANs
107
Work stations
Apps
106
105
Web Servers
104
UNIX Servers
103
Windows Servers
102
E-mail Servers
101

FIG. 1

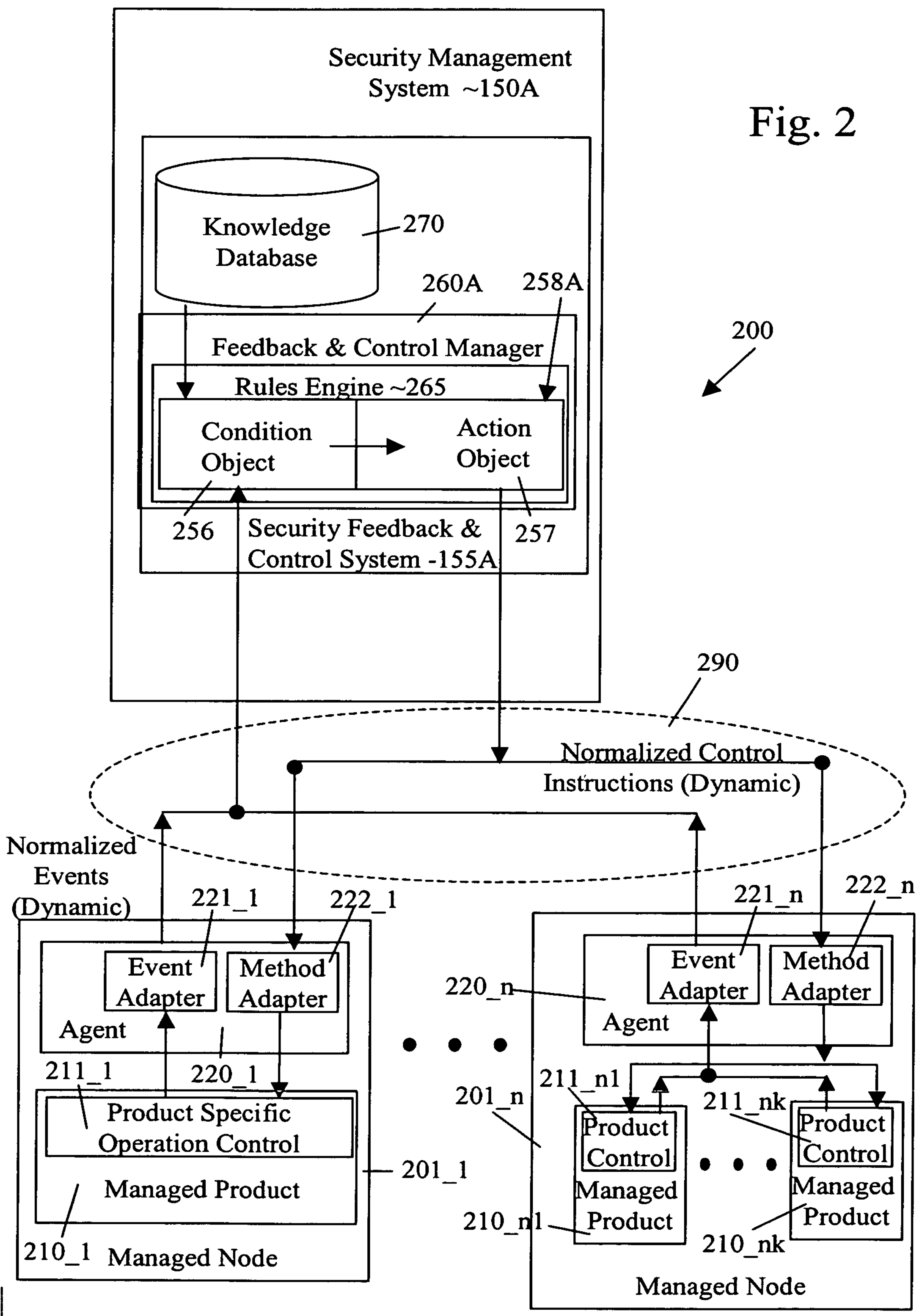

Security Management System ~150A
Knowledge Database
270
260A
258A
Feedback & Control Manager
Rules Engine ~265
Condition Object
Action Object
256
Security Feedback & Control System -155A
257
200
290
Normalized Control Instructions (Dynamic)
Normalized Events (Dynamic)
221_1
222_1
Event Adapter
Method Adapter
Agent
211_1
220_1
Product Specific Operation Control
Managed Product
201_1
210_1
Managed Node
221_n
222_n
220_n
Event Adapter
Method Adapter
Agent
211_n1
211_nk
201_n
Product Control
Managed Product
Product Control
Managed Product
210_n1
210_nk
Managed Node

Fig. 2

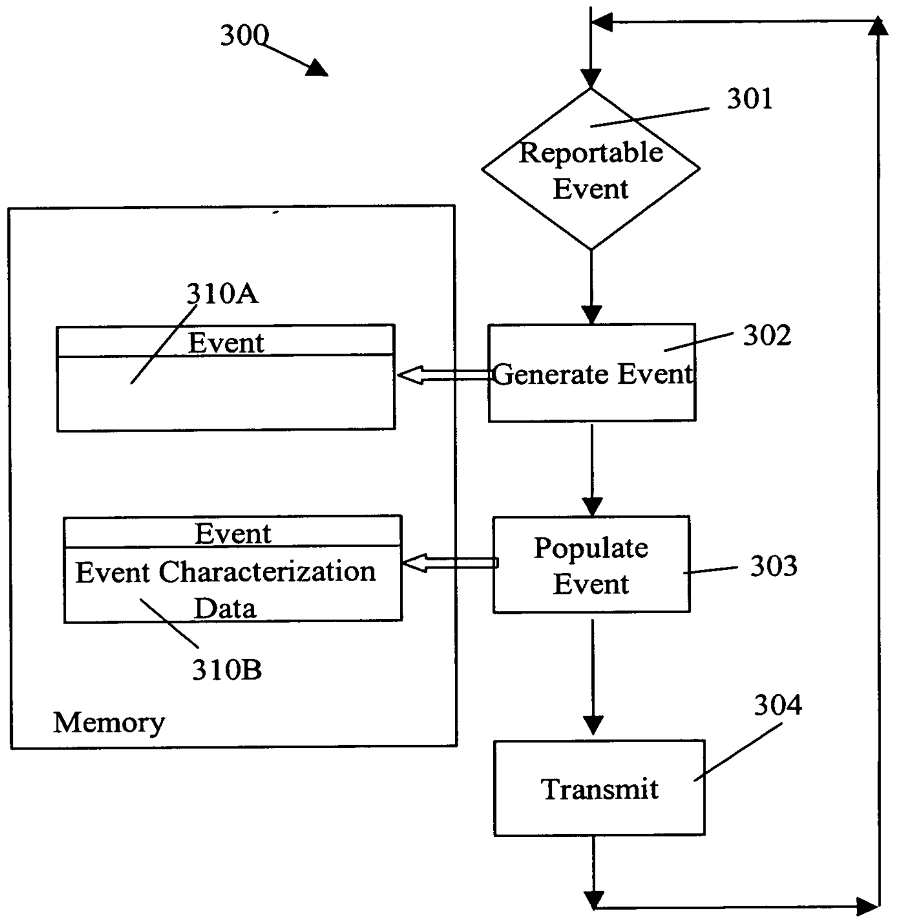

300
301
Reportable Event
302
Generate Event
303
Populate Event
304
Transmit
310A
Event
Event
Event Characterization Data
310B
Memory

Fig. 3

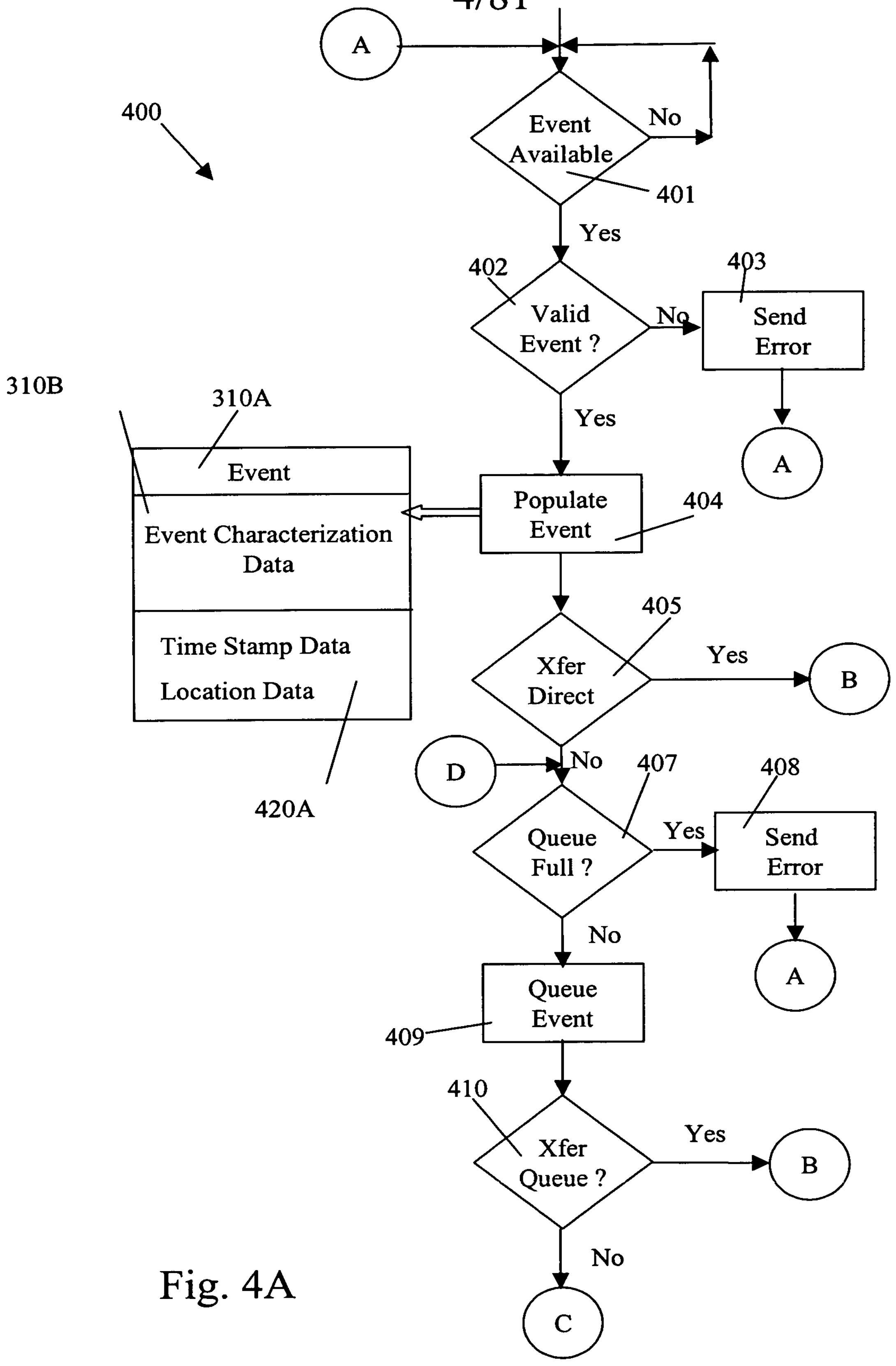

A
Event Available
No
401
Yes
402
Valid Event ?
No
403
Send Error
A
Yes
310B
310A
Event
Event Characterization Data
Time Stamp Data
Location Data
420A
Populate Event
404
405
Xfer Direct
Yes
B
D
No
407
Queue Full ?
Yes
408
Send Error
A
No
Queue Event
409
410
Xfer Queue ?
Yes
B
No
C
400

Fig. 4A

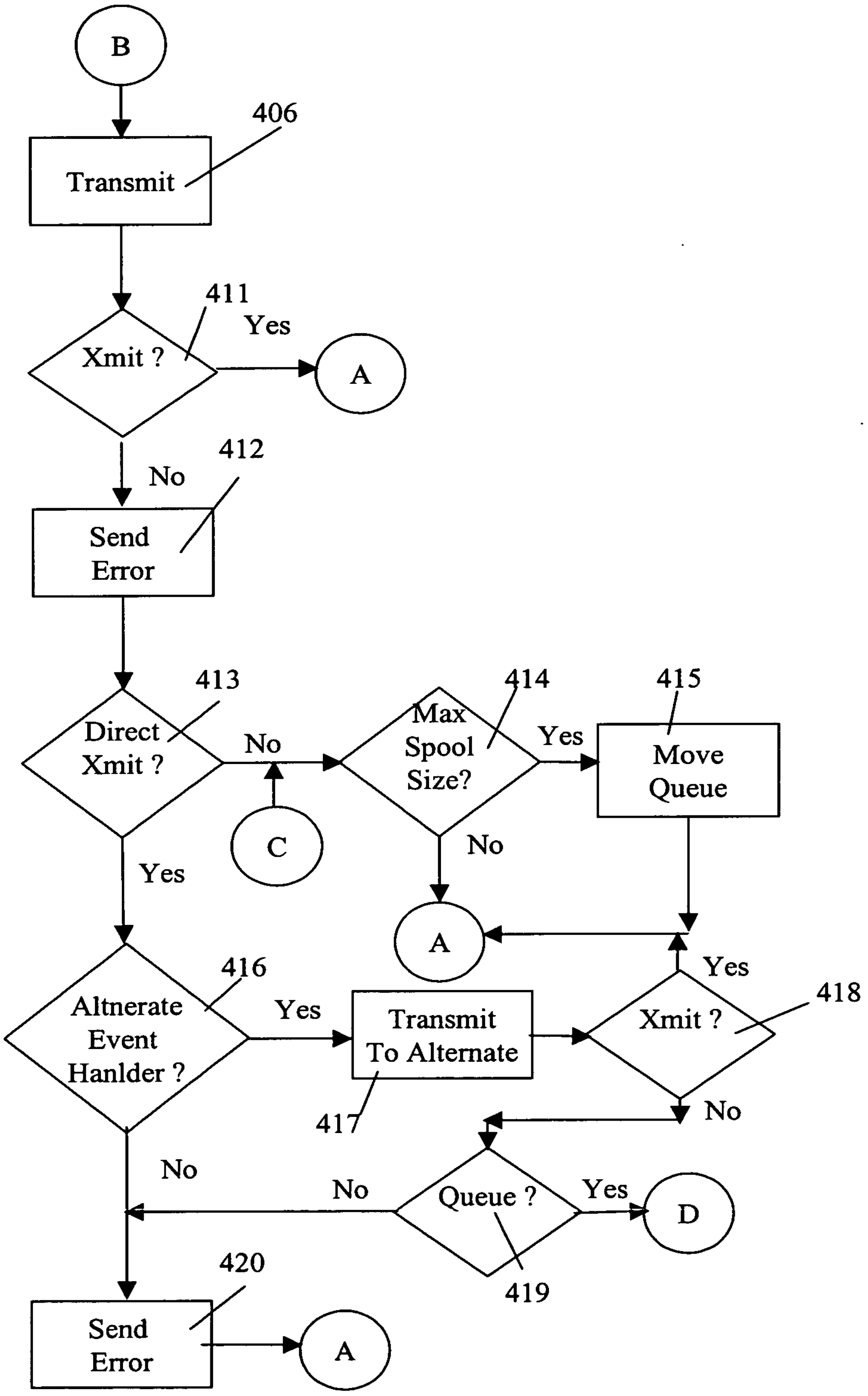

B
406
Transmit
411
Xmit ?
Yes
A
No
412
Send Error
413
Direct Xmit ?
No
C
414
Max Spool Size?
Yes
415
Move Queue
No
A
Yes
416
Altnerate Event Hanlder ?
Yes
Transmit To Alternate
417
418
Xmit ?
Yes
No
No
No
Queue ?
Yes
D
419
420
Send Error
A

Fig. 4B

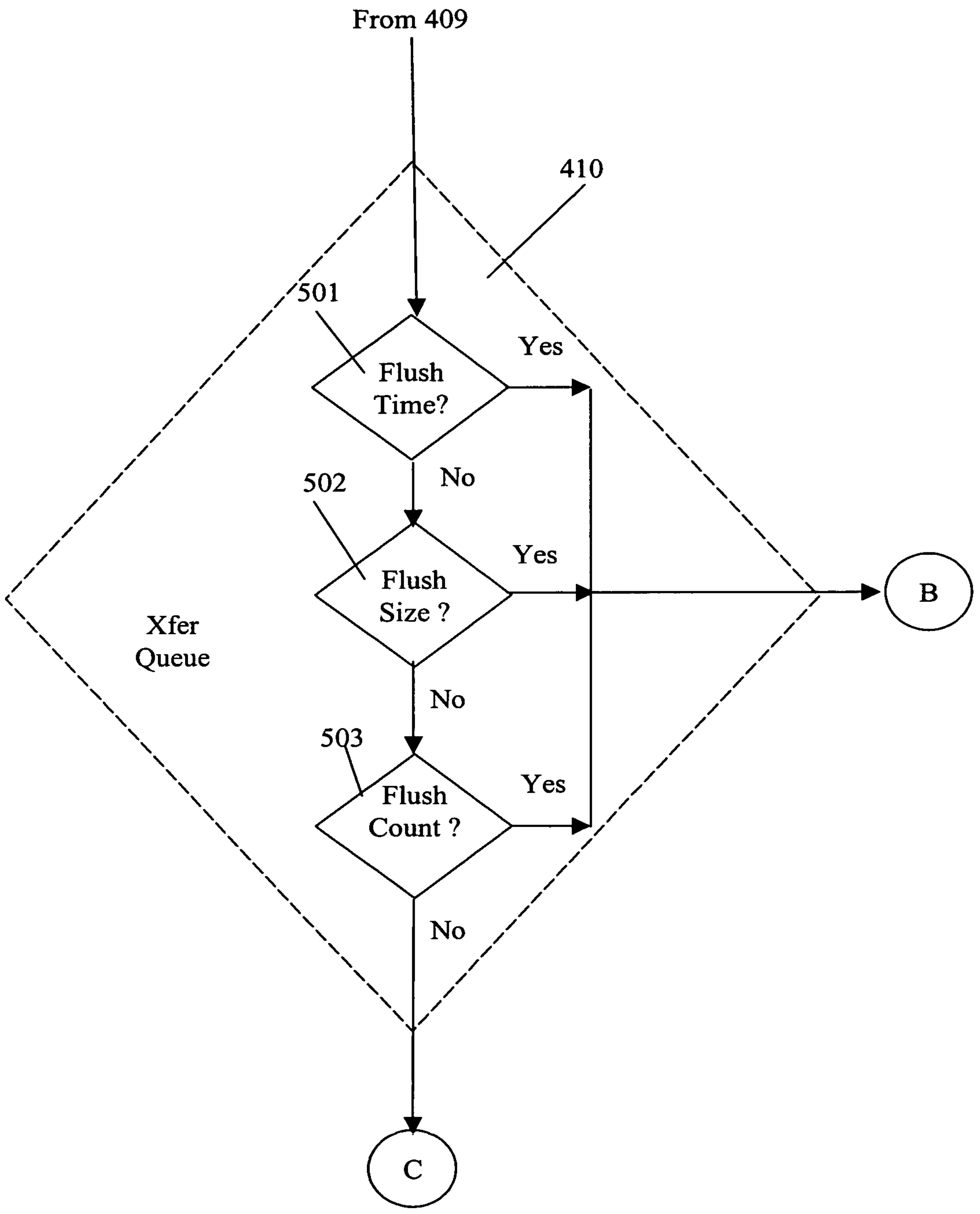

From 409
410
501
Flush Time?
Yes
No
502
Flush Size ?
Yes
No
503
Flush Count ?
Yes
No
Xfer Queue
B
C

Fig. 5A

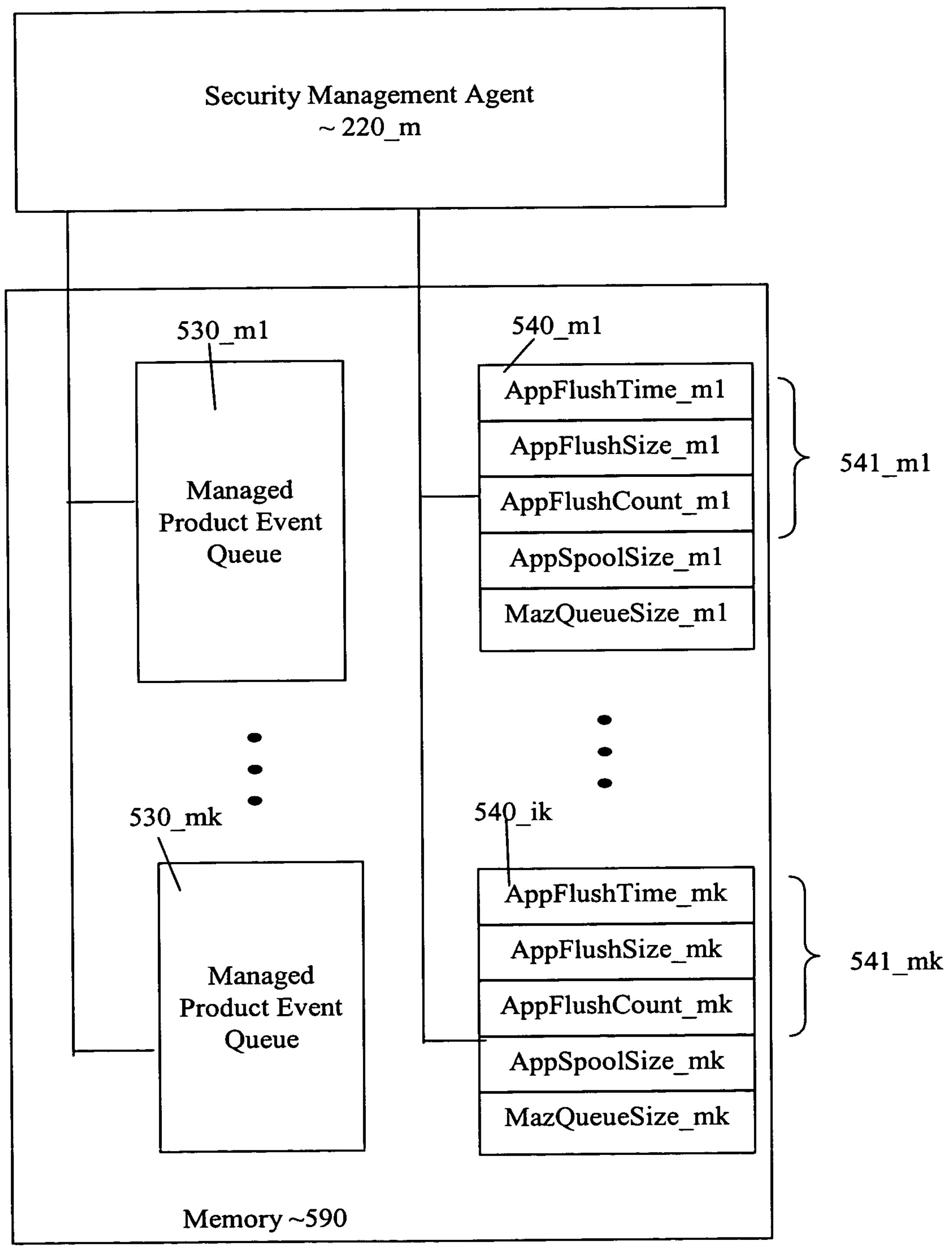

Security Management Agent
~ 220_m
530_m1
Managed Product Event Queue
530_mk
Managed Product Event Queue
540_m1
AppFlushTime_m1
AppFlushSize_m1
AppFlushCount_m1
AppSpoolSize_m1
MazQueueSize_m1
541_m1
540_ik
AppFlushTime_mk
AppFlushSize_mk
AppFlushCount_mk
AppSpoolSize_mk
MazQueueSize_mk
541_mk
Memory ~590

Fig. 5B

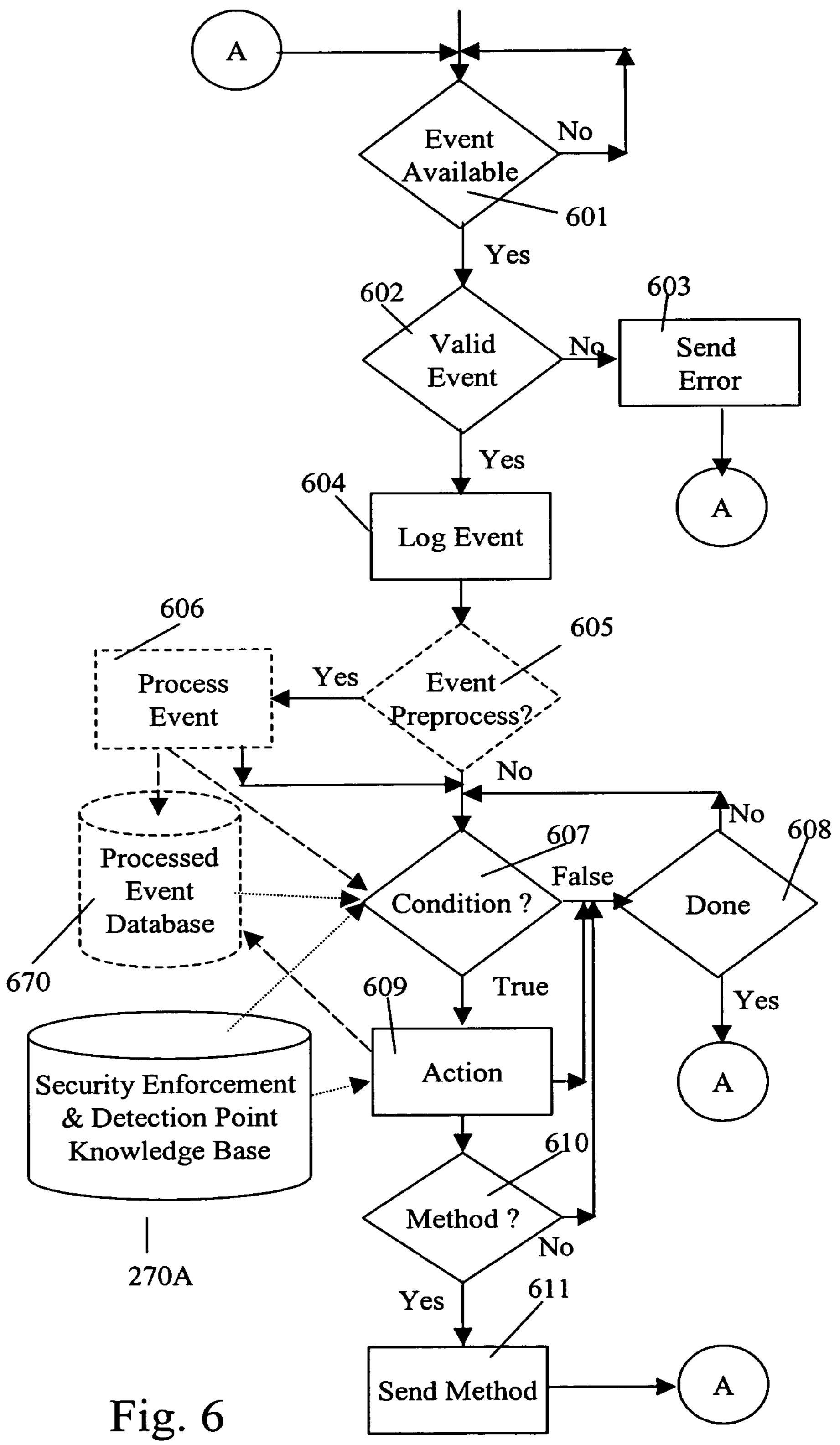

A
Event Available
No
601
Yes
602
Valid Event
No
603
Send Error
A
Yes
604
Log Event
606
605
Process Event
Yes
Event Preprocess?
No
Processed Event Database
670
Condition ?
607
False
No
608
Done
True
609
Yes
A
Security Enforcement & Detection Point Knowledge Base
Action
270A
610
Method ?
No
Yes
611
Send Method
A

Fig. 6

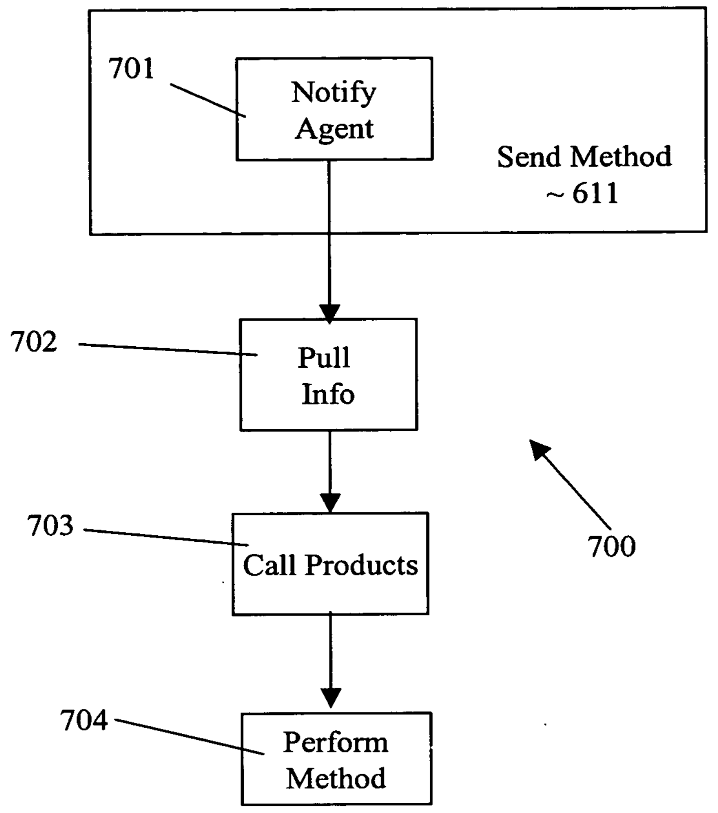

701
Notify Agent
Send Method ~ 611
702
Pull Info
703
Call Products
704
Perform Method
700

Fig. 7

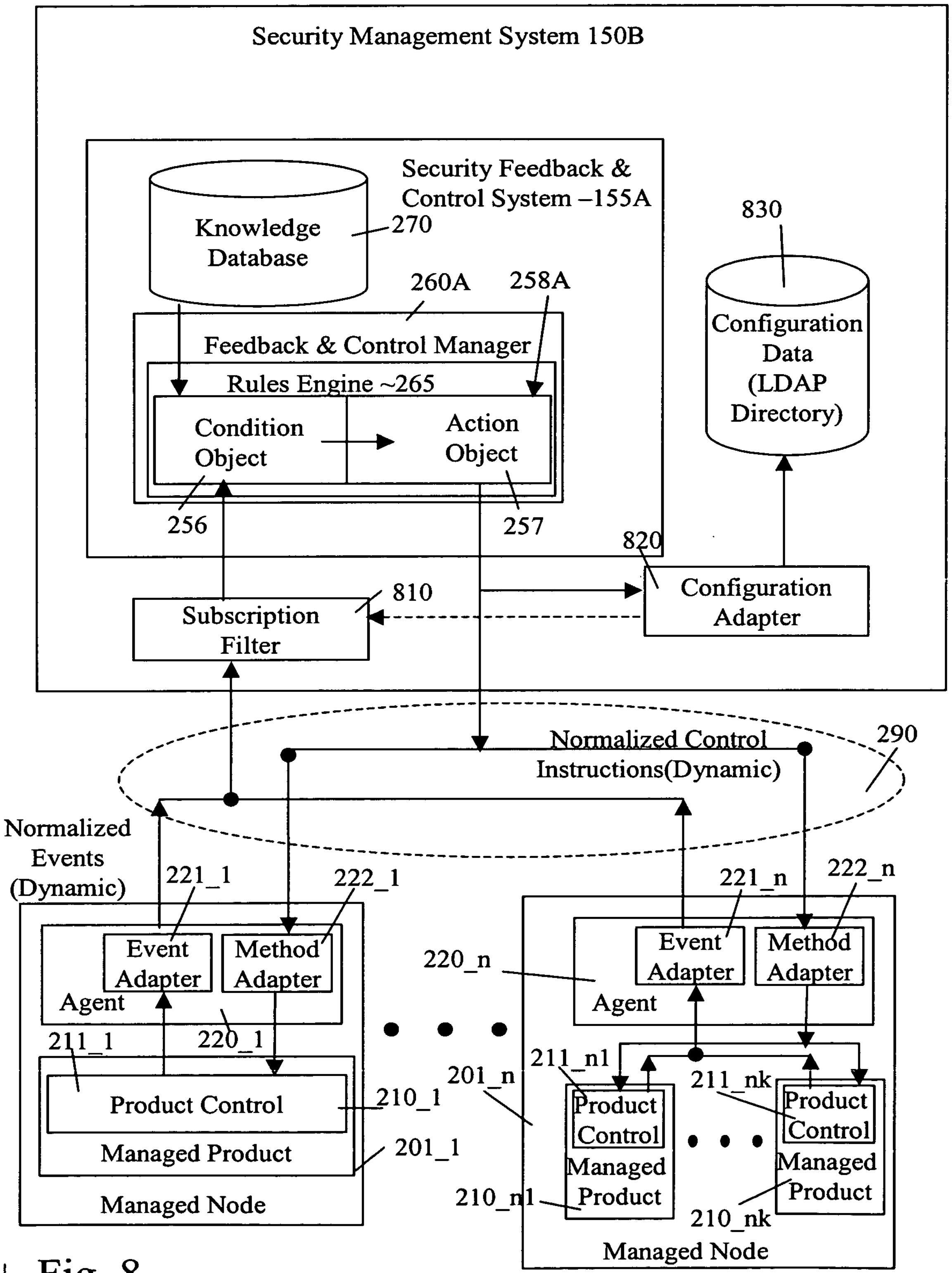

Security Management System 150B
Security Feedback & Control System –155A
Knowledge Database
270
260A
258A
Feedback & Control Manager
Rules Engine ~265
Condition Object
Action Object
256
257
830
Configuration Data (LDAP Directory)
820
Configuration Adapter
810
Subscription Filter
Normalized Control Instructions(Dynamic)
290
Normalized Events (Dynamic)
221_1
222_1
Event Adapter
Method Adapter
Agent
211_1
220_1
Product Control
210_1
Managed Product
201_1
Managed Node
220_n
221_n
222_n
Event Adapter
Method Adapter
Agent
211_n1
211_nk
201_n
Product Control
Managed Product
210_n1
210_nk
Managed Node

Fig. 8

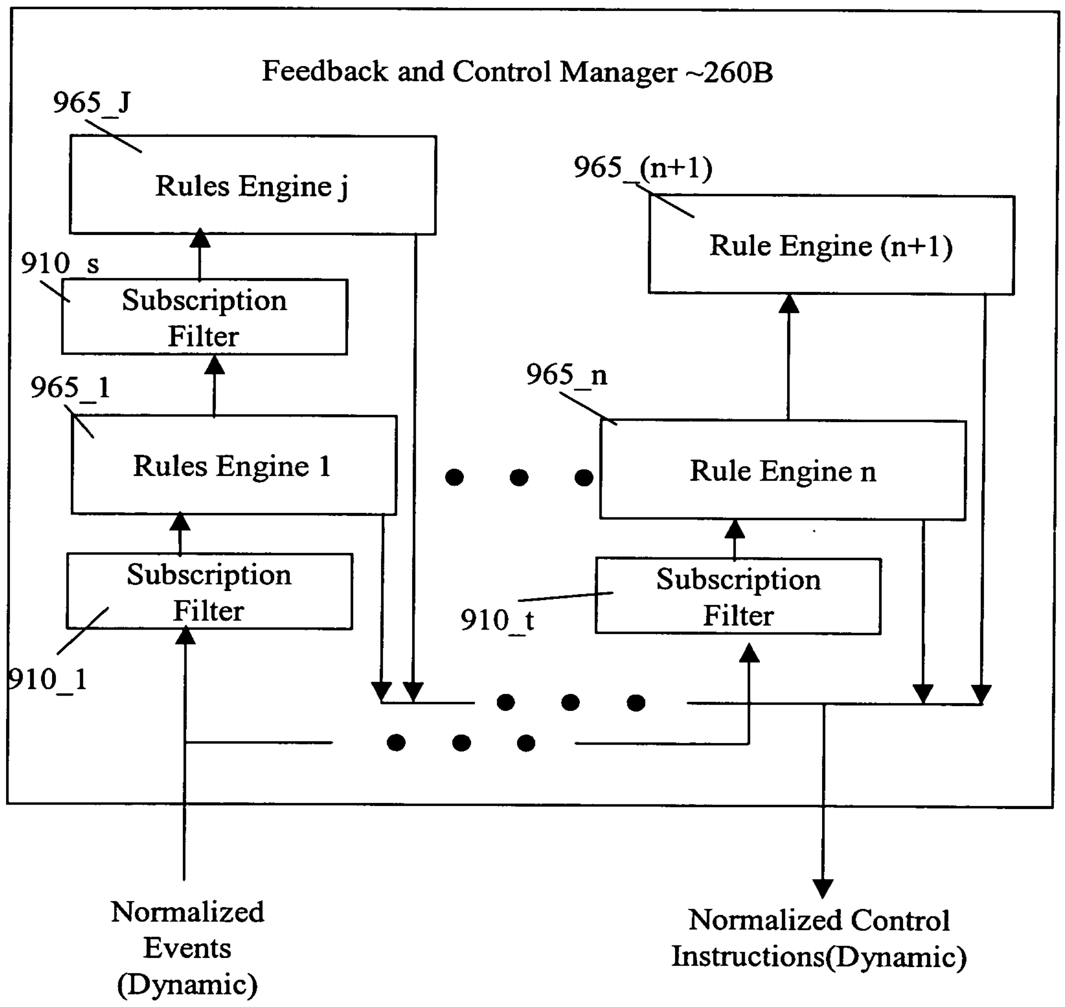

Feedback and Control Manager ~260B
965_J
Rules Engine j
965_(n+1)
Rule Engine (n+1)
910_s
Subscription Filter
965_1
Rules Engine 1
965_n
Rule Engine n
Subscription Filter
910_t
Subscription Filter
910_1
Normalized Events (Dynamic)
Normalized Control Instructions(Dynamic)

Fig. 9

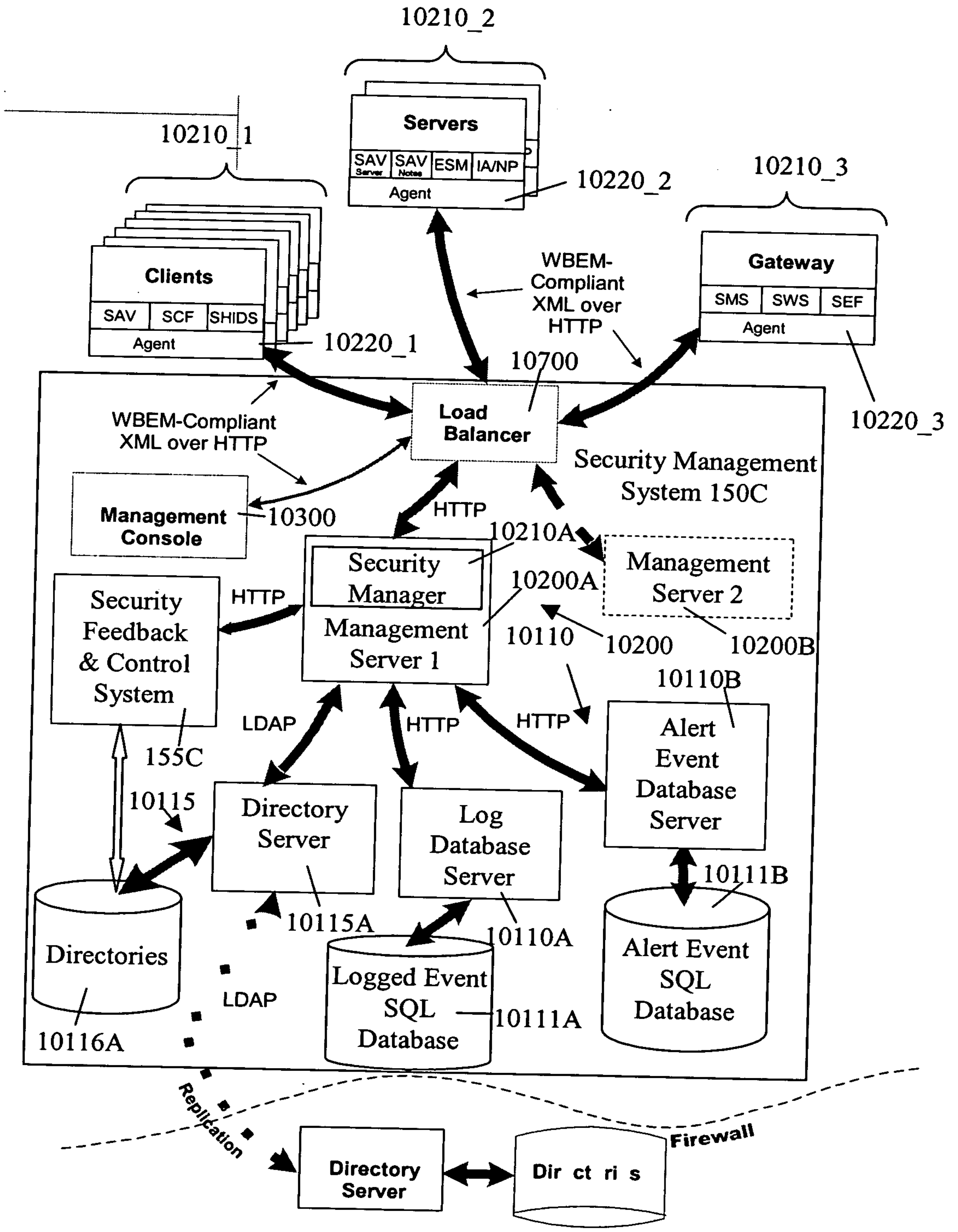

10210_2
Servers
SAV Server
SAV Notes
ESM
IA/NP
Agent
10220_2
10210_1
Clients
SAV
SCF
SHIDS
Agent
10220_1
10210_3
Gateway
SMS
SWS
SEF
Agent
10220_3
WBEM-Compliant XML over HTTP
10700
Load Balancer
WBEM-Compliant XML over HTTP
Security Management System 150C
Management Console
10300
HTTP
10210A
Security Manager
Management Server 1
10200A
Management Server 2
10110
10200
10200B
Security Feedback & Control System
HTTP
10110B
LDAP
HTTP
HTTP
155C
10115
Directory Server
Log Database Server
Alert Event Database Server
10111B
Directories
10115A
10110A
Logged Event SQL Database
10111A
Alert Event SQL Database
10116A
LDAP
Replication
Firewall
Directory Server
Dir ct ri s

Fig. 10

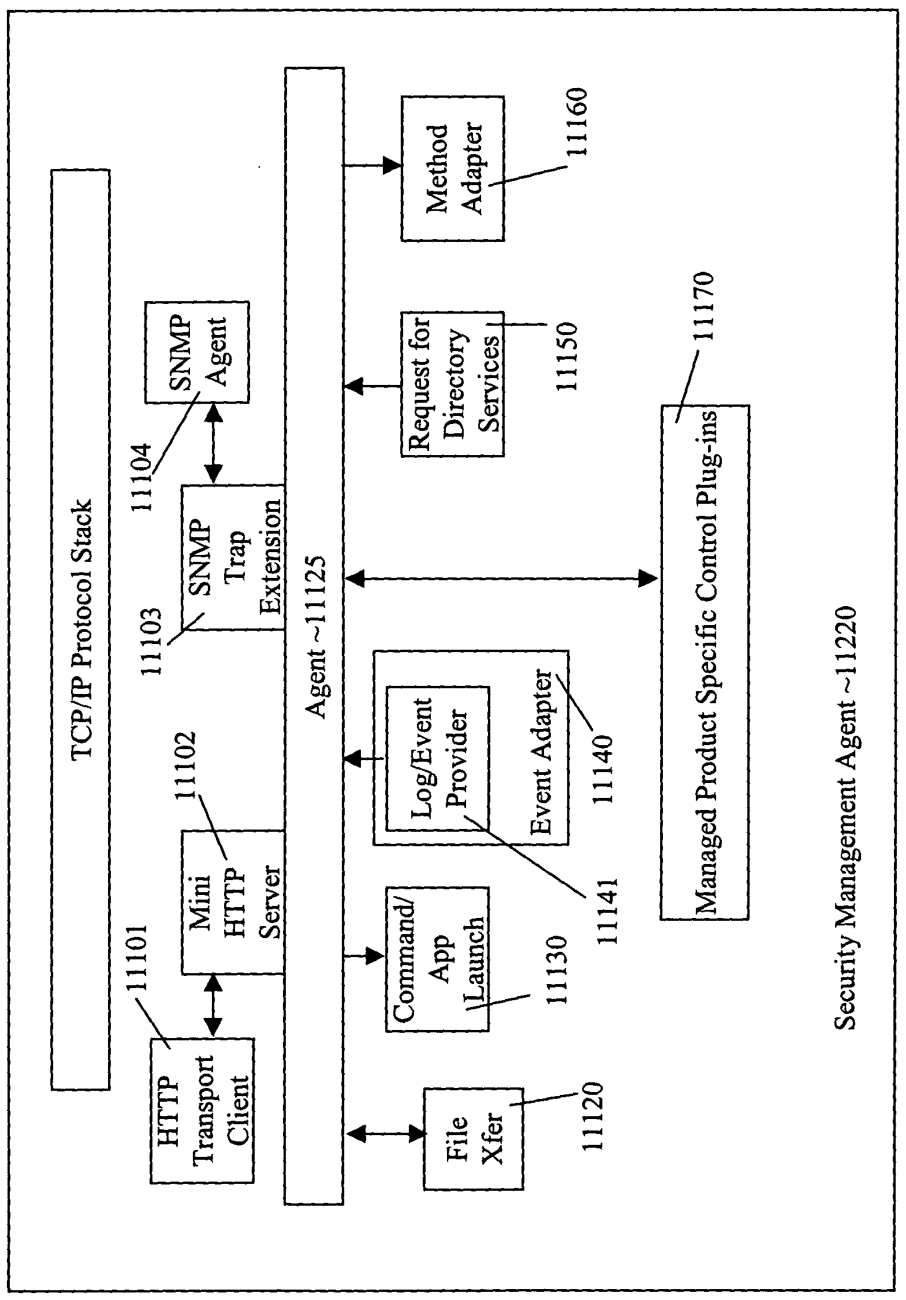

TCP/IP Protocol Stack
HTTP Transport Client
11101
Mini HTTP Server
11102
SNMP Trap Extension
11103
SNMP Agent
11104
Agent ~11125
File Xfer
11120
Command/ App Launch
11130
Log/Event Provider
11141
Event Adapter
11140
Request for Directory Services
11150
Method Adapter
11160
Managed Product Specific Control Plug-ins
11170
Security Management Agent ~11220

FIG. 11

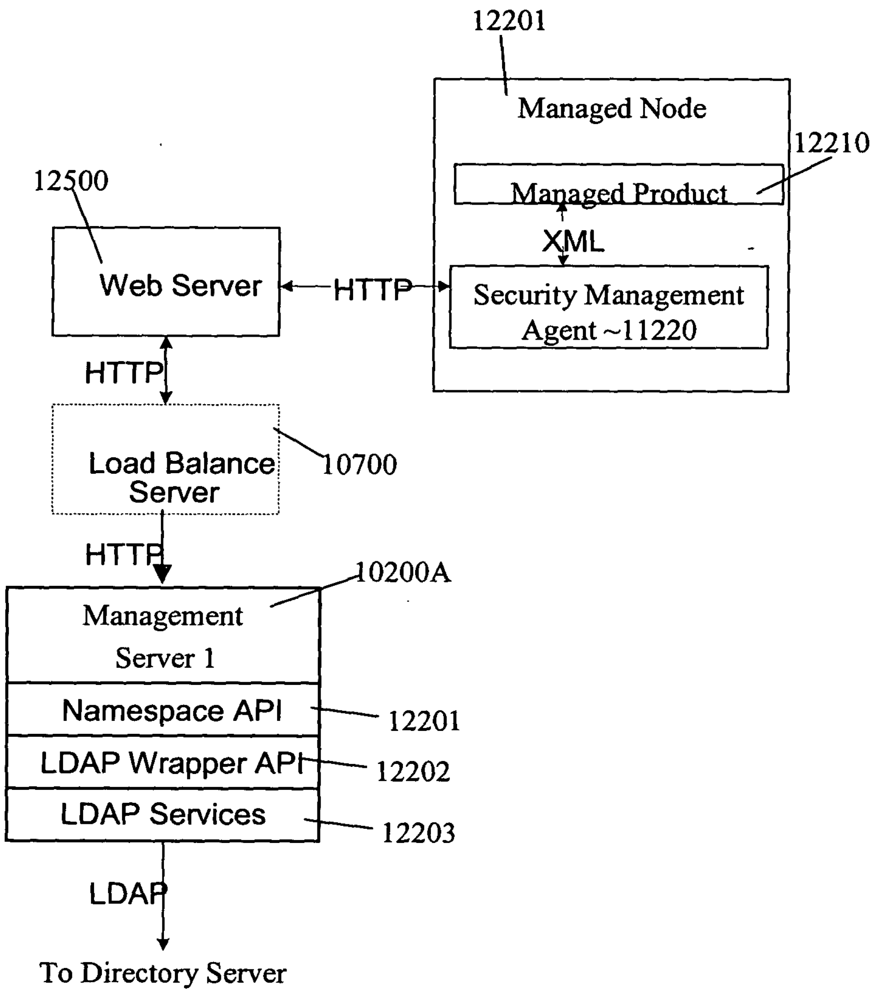

12201
Managed Node
12210
Managed Product
XML
Security Management Agent ~11220
12500
Web Server
HTTP
HTTP
Load Balance Server
10700
HTTP
10200A
Management Server 1
Namespace API
12201
LDAP Wrapper API
12202
LDAP Services
12203
LDAP
To Directory Server

Fig. 12A

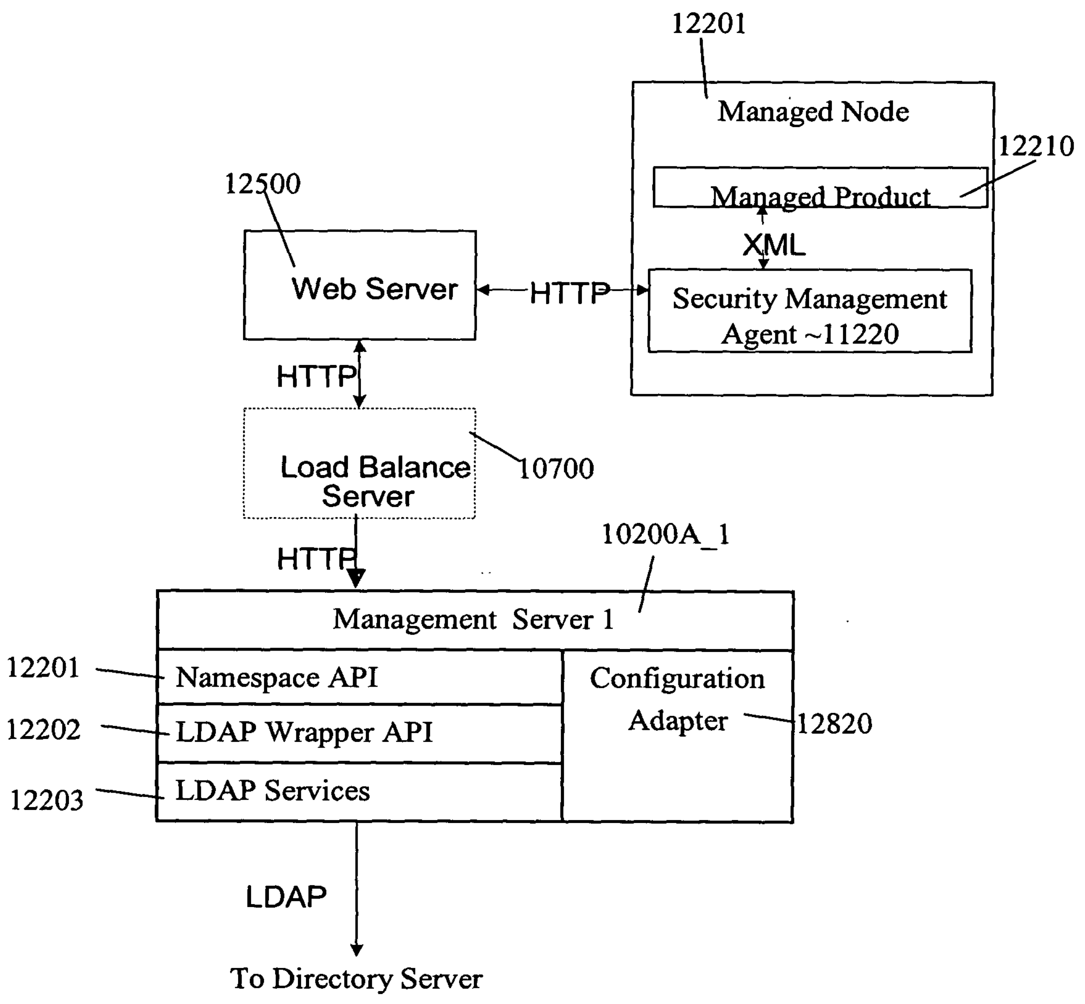

12201
Managed Node
12210
Managed Product
XML
Security Management Agent ~11220
12500
Web Server
HTTP
HTTP
Load Balance Server
10700
HTTP
10200A_1
Management Server 1
12201
Namespace API
12202
LDAP Wrapper API
12203
LDAP Services
Configuration Adapter
12820
LDAP
To Directory Server

Fig. 12B

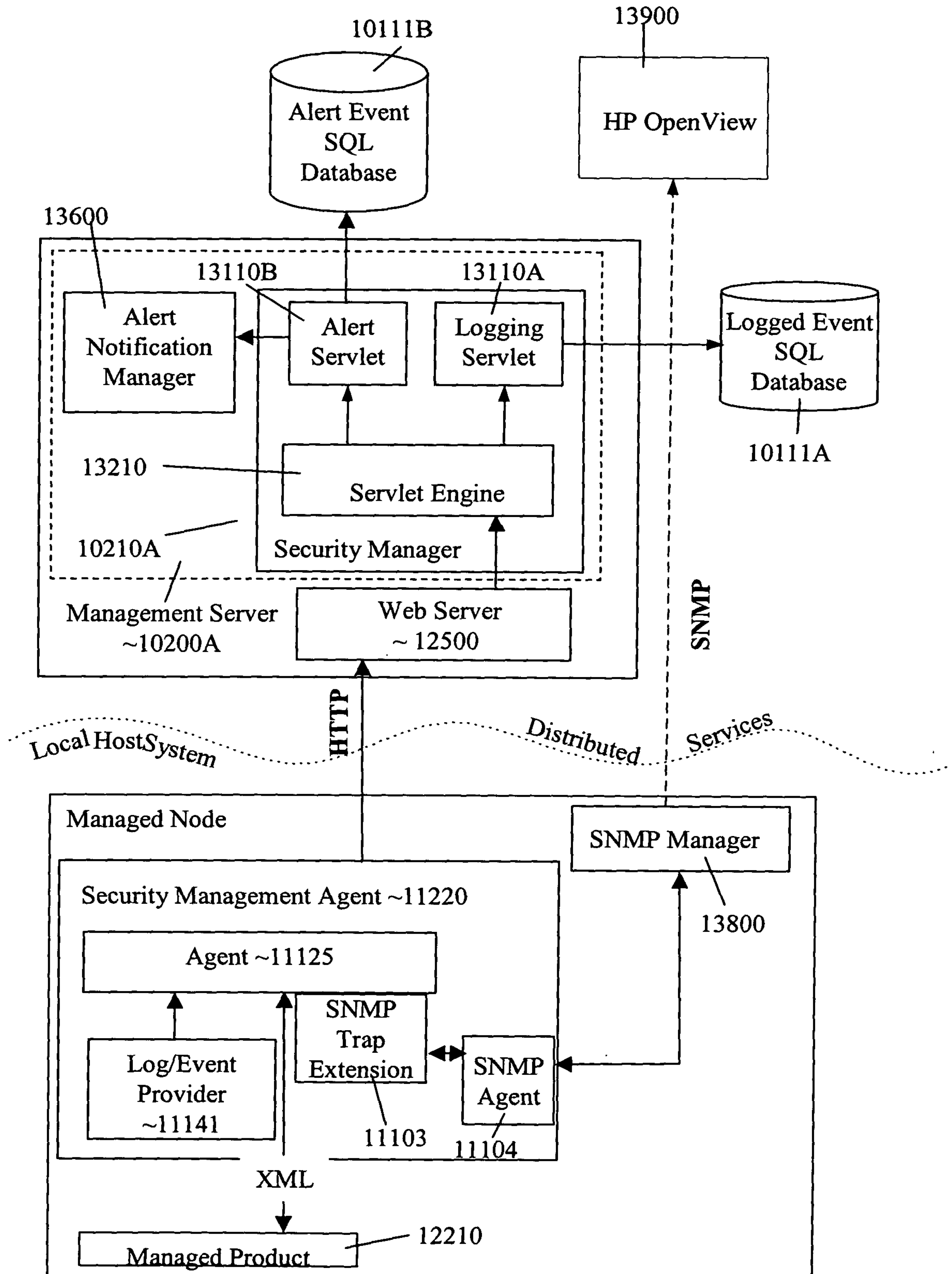

10111B
Alert Event SQL Database
13900
HP OpenView
13600
13110B
13110A
Alert Notification Manager
Alert Servlet
Logging Servlet
Logged Event SQL Database
10111A
13210
Servlet Engine
10210A
Security Manager
Management Server ~10200A
Web Server ~ 12500
SNMP
HTTP
Local HostSystem
Distributed Services
Managed Node
SNMP Manager
Security Management Agent ~11220
13800
Agent ~11125
SNMP Trap Extension
Log/Event Provider ~11141
SNMP Agent
11103
11104
XML
Managed Product
12210

Fig. 13

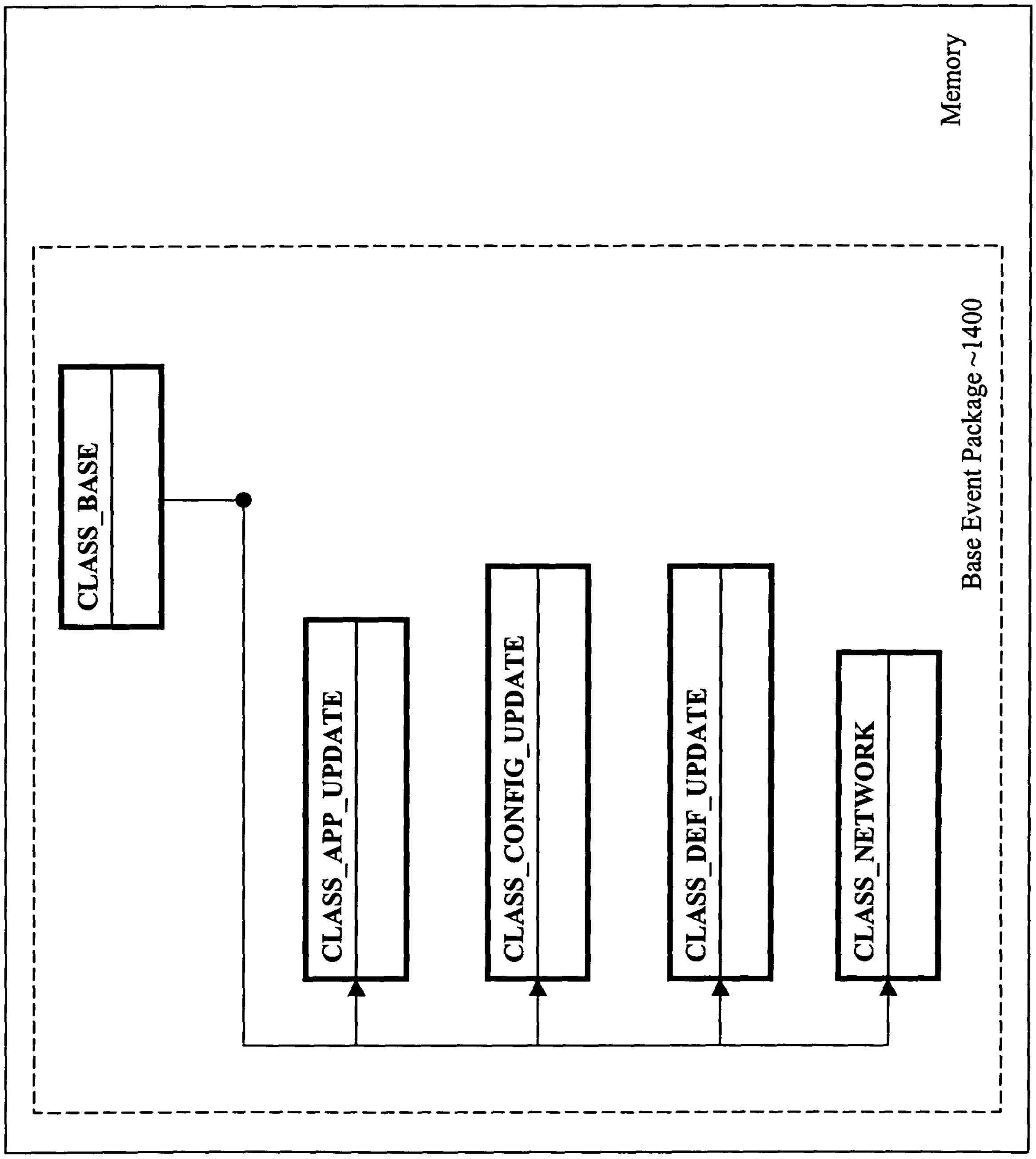

Memory
Base Event Package ~1400
CLASS_BASE
CLASS_APP_UPDATE
CLASS_CONFIG_UPDATE
CLASS_DEF_UPDATE
CLASS_NETWORK

FIG. 14

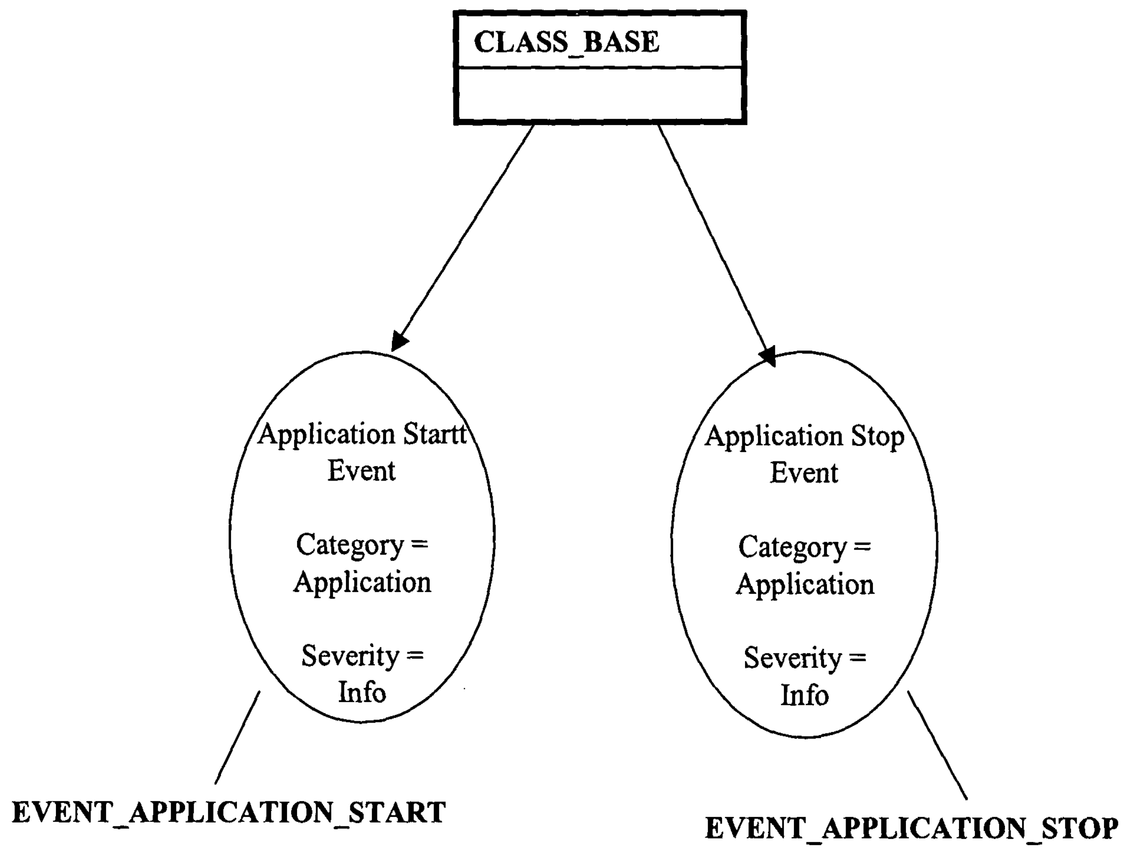
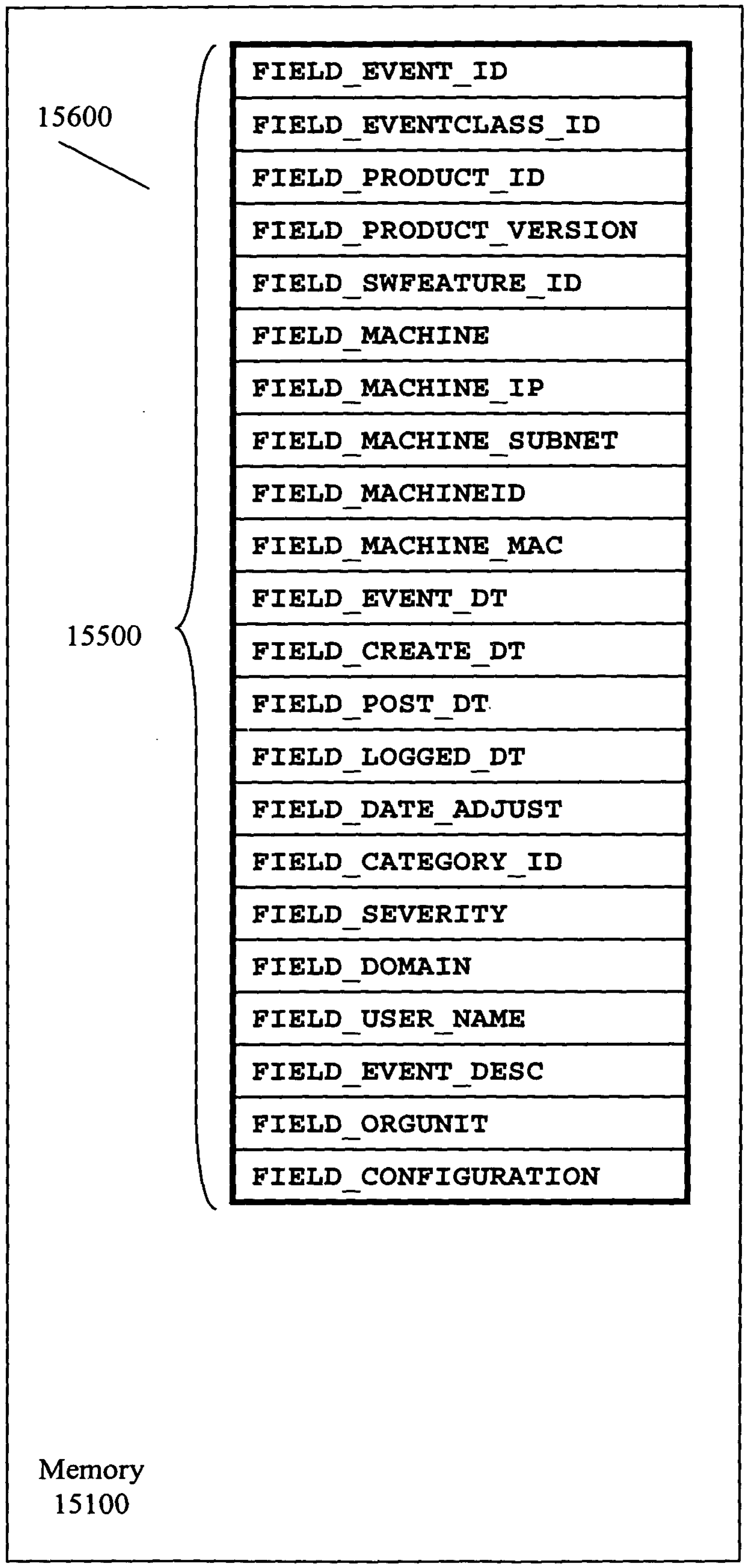

CLASS_BASE
Application Startt Event
Category = Application
Severity = Info
Application Stop Event
Category = Application
Severity = Info
EVENT_APPLICATION_START
EVENT_APPLICATION_STOP

FIG. 15A



15600
15500
FIELD_EVENT_ID
FIELD_EVENTCLASS_ID
FIELD_PRODUCT_ID
FIELD_PRODUCT_VERSION
FIELD_SWFEATURE_ID
FIELD_MACHINE
FIELD_MACHINE_IP
FIELD_MACHINE_SUBNET
FIELD_MACHINEID
FIELD_MACHINE_MAC
FIELD_EVENT_DT
FIELD_CREATE_DT
FIELD_POST_DT
FIELD_LOGGED_DT
FIELD_DATE_ADJUST
FIELD_CATEGORY_ID
FIELD_SEVERITY
FIELD_DOMAIN
FIELD_USER_NAME
FIELD_EVENT_DESC
FIELD_ORGUNIT
FIELD_CONFIGURATION
Memory
15100

Fig. 15B

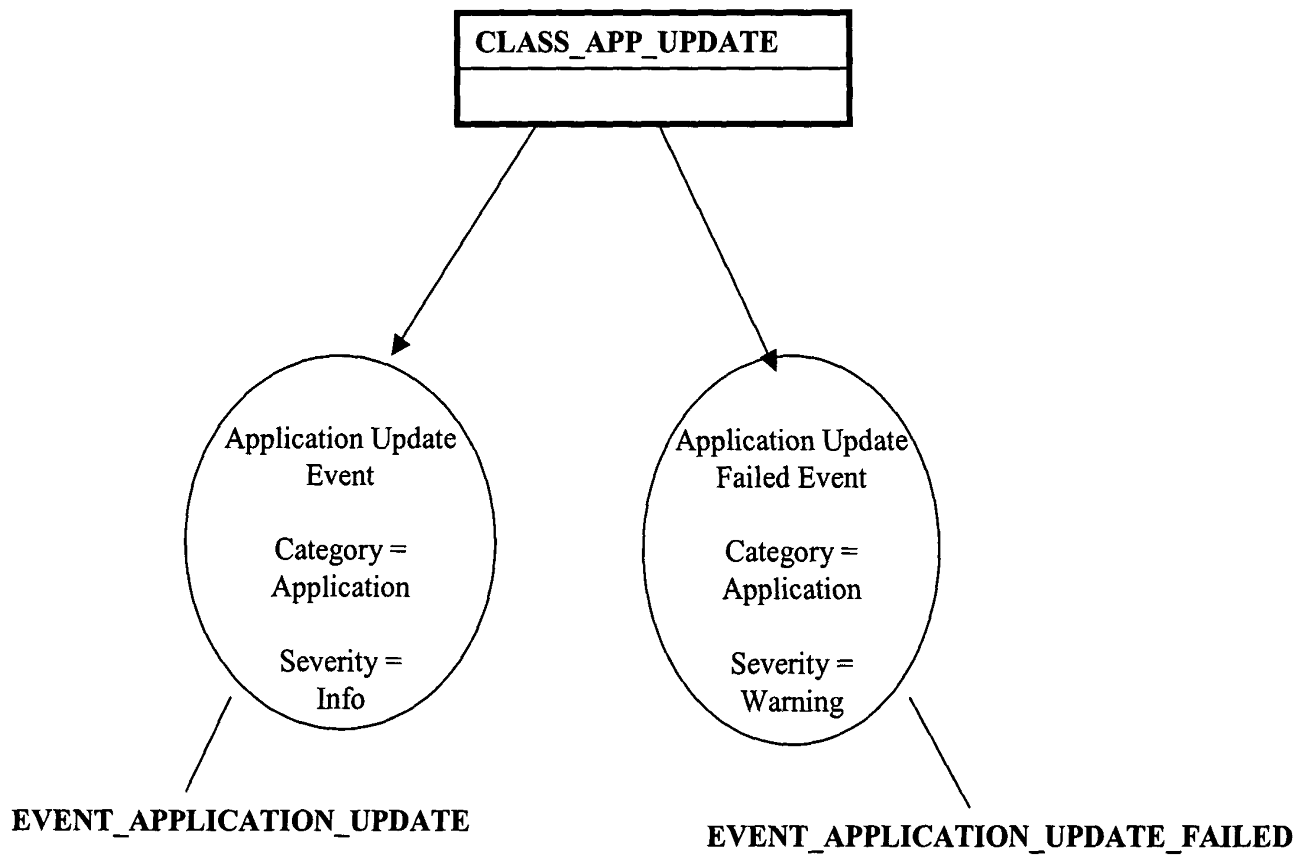
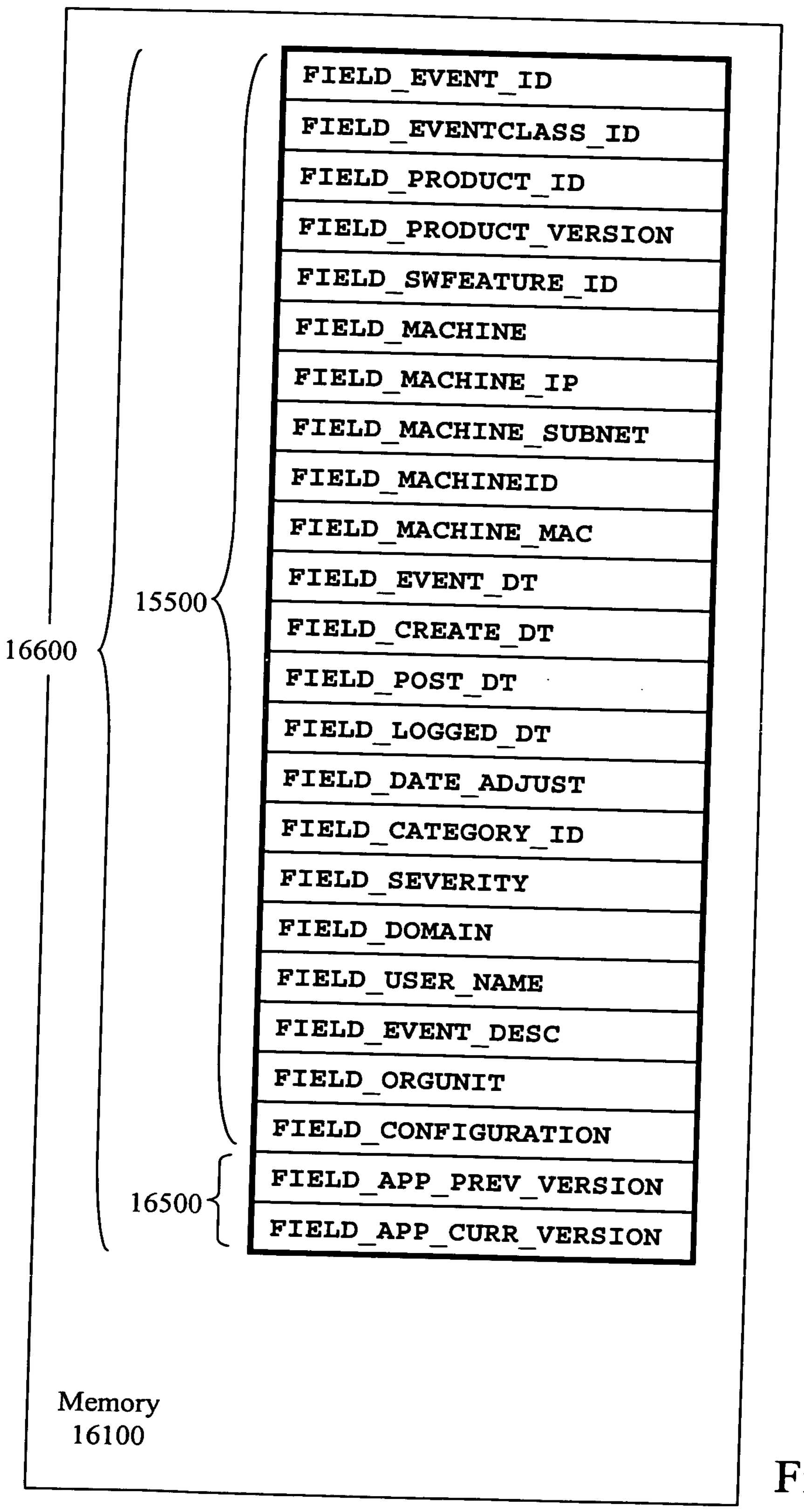
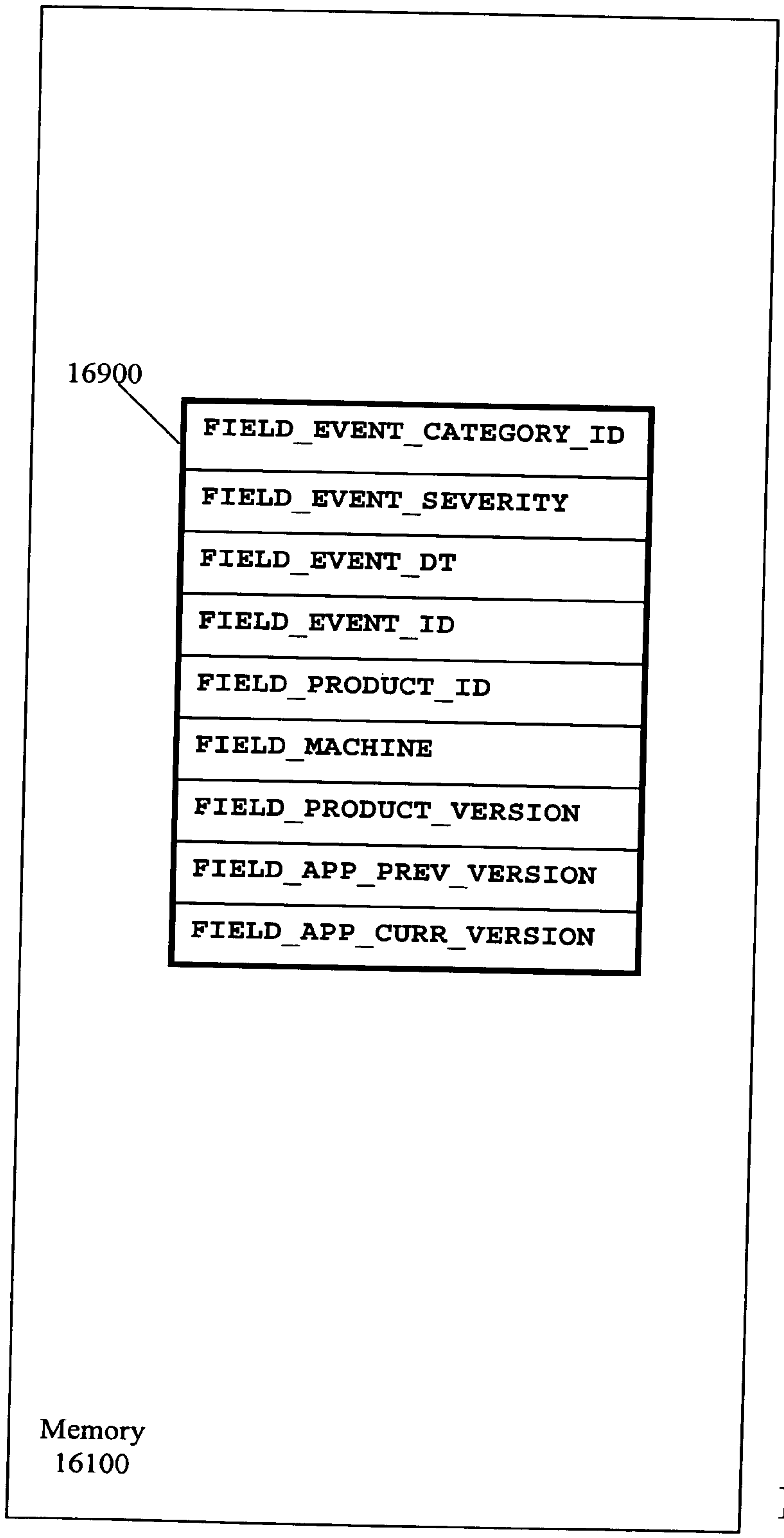

CLASS_APP_UPDATE
Application Update Event
Category = Application
Severity = Info
Application Update Failed Event
Category = Application
Severity = Warning
EVENT_APPLICATION_UPDATE
EVENT_APPLICATION_UPDATE_FAILED

FIG. 16A



FIELD_EVENT_ID
FIELD_EVENTCLASS_ID
FIELD_PRODUCT_ID
FIELD_PRODUCT_VERSION
FIELD_SWFEATURE_ID
FIELD_MACHINE
FIELD_MACHINE_IP
FIELD_MACHINE_SUBNET
FIELD_MACHINEID
FIELD_MACHINE_MAC
FIELD_EVENT_DT
FIELD_CREATE_DT
FIELD_POST_DT
FIELD_LOGGED_DT
FIELD_DATE_ADJUST
FIELD_CATEGORY_ID
FIELD_SEVERITY
FIELD_DOMAIN
FIELD_USER_NAME
FIELD_EVENT_DESC
FIELD_ORGUNIT
FIELD_CONFIGURATION
FIELD_APP_PREV_VERSION
FIELD_APP_CURR_VERSION
15500
16600
16500
Memory
16100

Fig. 16B



16900
FIELD_EVENT_CATEGORY_ID
FIELD_EVENT_SEVERITY
FIELD_EVENT_DT
FIELD_EVENT_ID
FIELD_PRODUCT_ID
FIELD_MACHINE
FIELD_PRODUCT_VERSION
FIELD_APP_PREV_VERSION
FIELD_APP_CURR_VERSION
Memory
16100

Fig. 16C

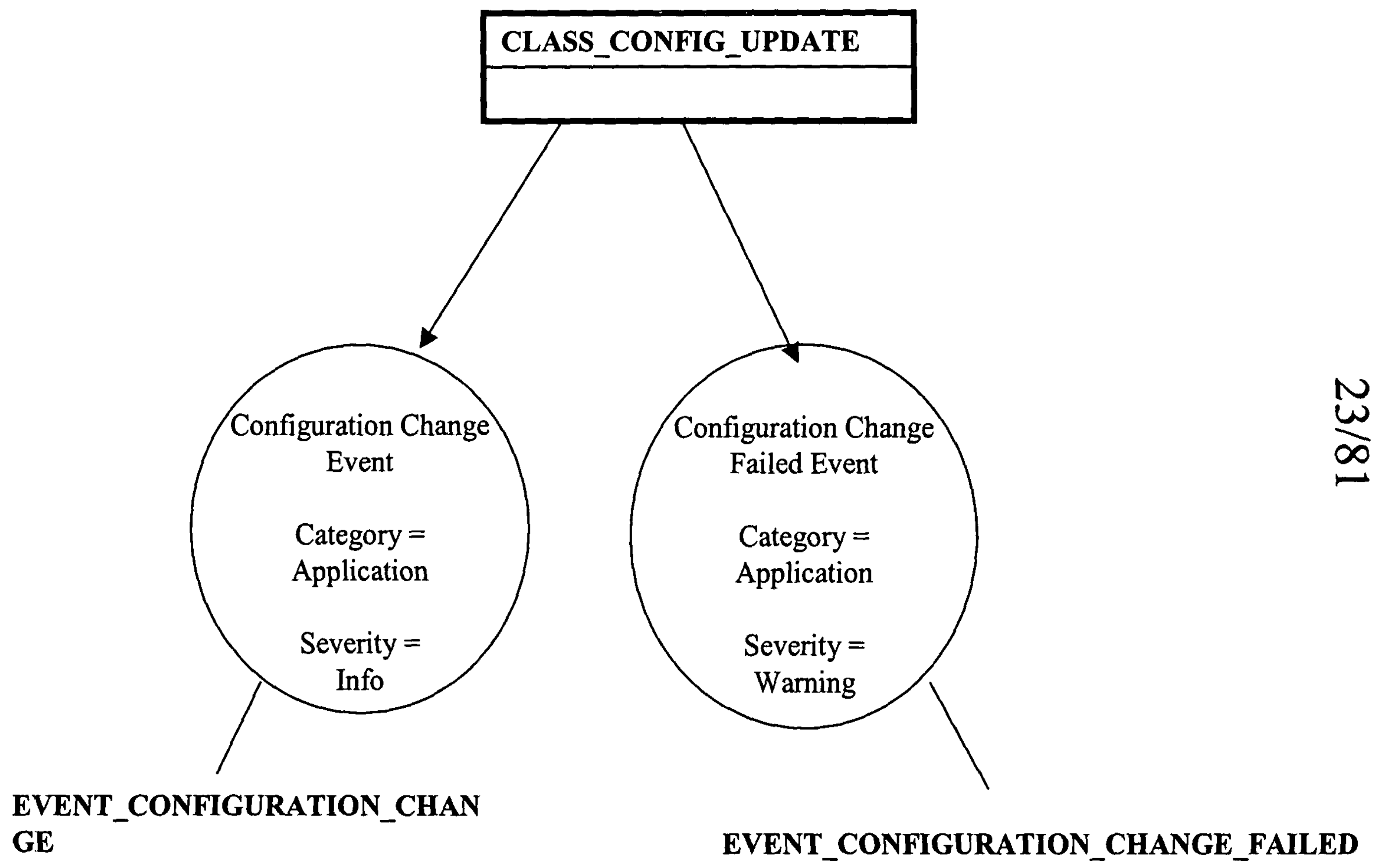
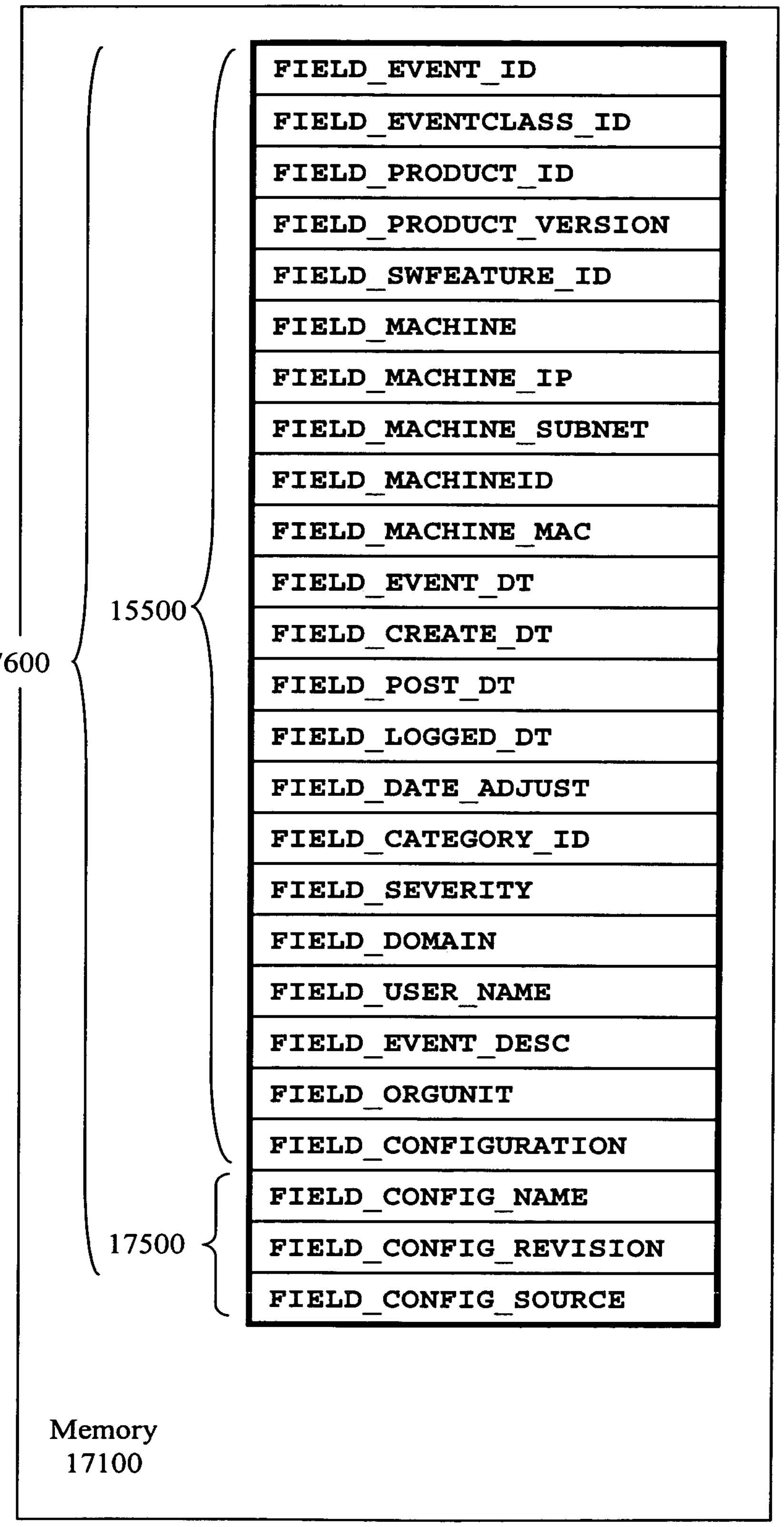

CLASS_CONFIG_UPDATE
Configuration Change Event
Category = Application
Severity = Info
Configuration Change Failed Event
Category = Application
Severity = Warning
EVENT_CONFIGURATION_CHANGE
EVENT_CONFIGURATION_CHANGE_FAILED

FIG. 17A



FIELD_EVENT_ID
FIELD_EVENTCLASS_ID
FIELD_PRODUCT_ID
FIELD_PRODUCT_VERSION
FIELD_SWFEATURE_ID
FIELD_MACHINE
FIELD_MACHINE_IP
FIELD_MACHINE_SUBNET
FIELD_MACHINEID
FIELD_MACHINE_MAC
FIELD_EVENT_DT
FIELD_CREATE_DT
FIELD_POST_DT
FIELD_LOGGED_DT
FIELD_DATE_ADJUST
FIELD_CATEGORY_ID
FIELD_SEVERITY
FIELD_DOMAIN
FIELD_USER_NAME
FIELD_EVENT_DESC
FIELD_ORGUNIT
FIELD_CONFIGURATION
FIELD_CONFIG_NAME
FIELD_CONFIG_REVISION
FIELD_CONFIG_SOURCE
15500
17600
17500
Memory
17100

Fig. 17B

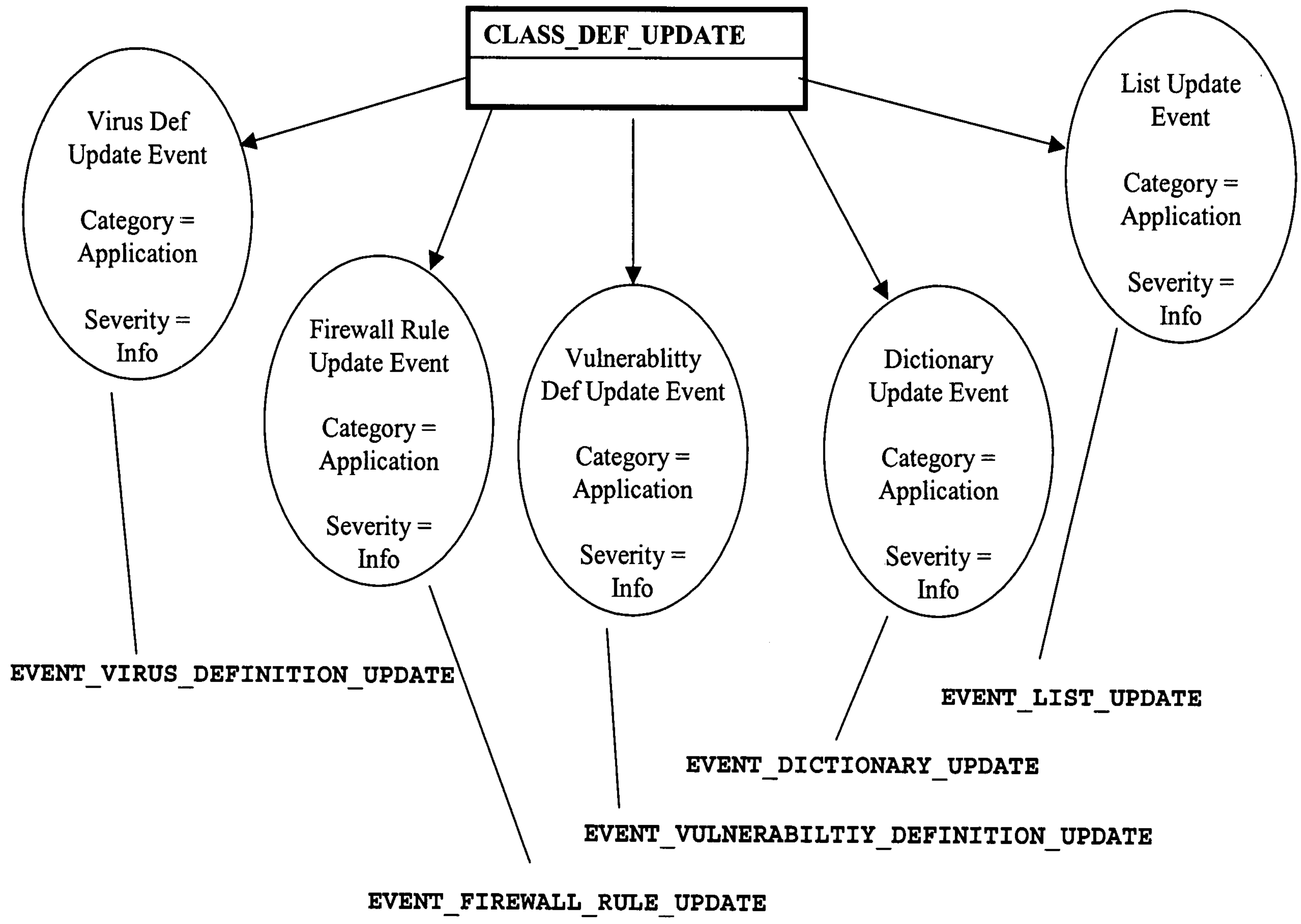

CLASS_DEF_UPDATE
Virus Def Update Event
Category = Application
Severity = Info
Firewall Rule Update Event
Category = Application
Severity = Info
Vulnerablitty Def Update Event
Category = Application
Severity = Info
Dictionary Update Event
Category = Application
Severity = Info
List Update Event
Category = Application
Severity = Info
EVENT_VIRUS_DEFINITION_UPDATE
EVENT_FIREWALL_RULE_UPDATE
EVENT_VULNERABILTIY_DEFINITION_UPDATE
EVENT_DICTIONARY_UPDATE
EVENT_LIST_UPDATE

FIG. 18A

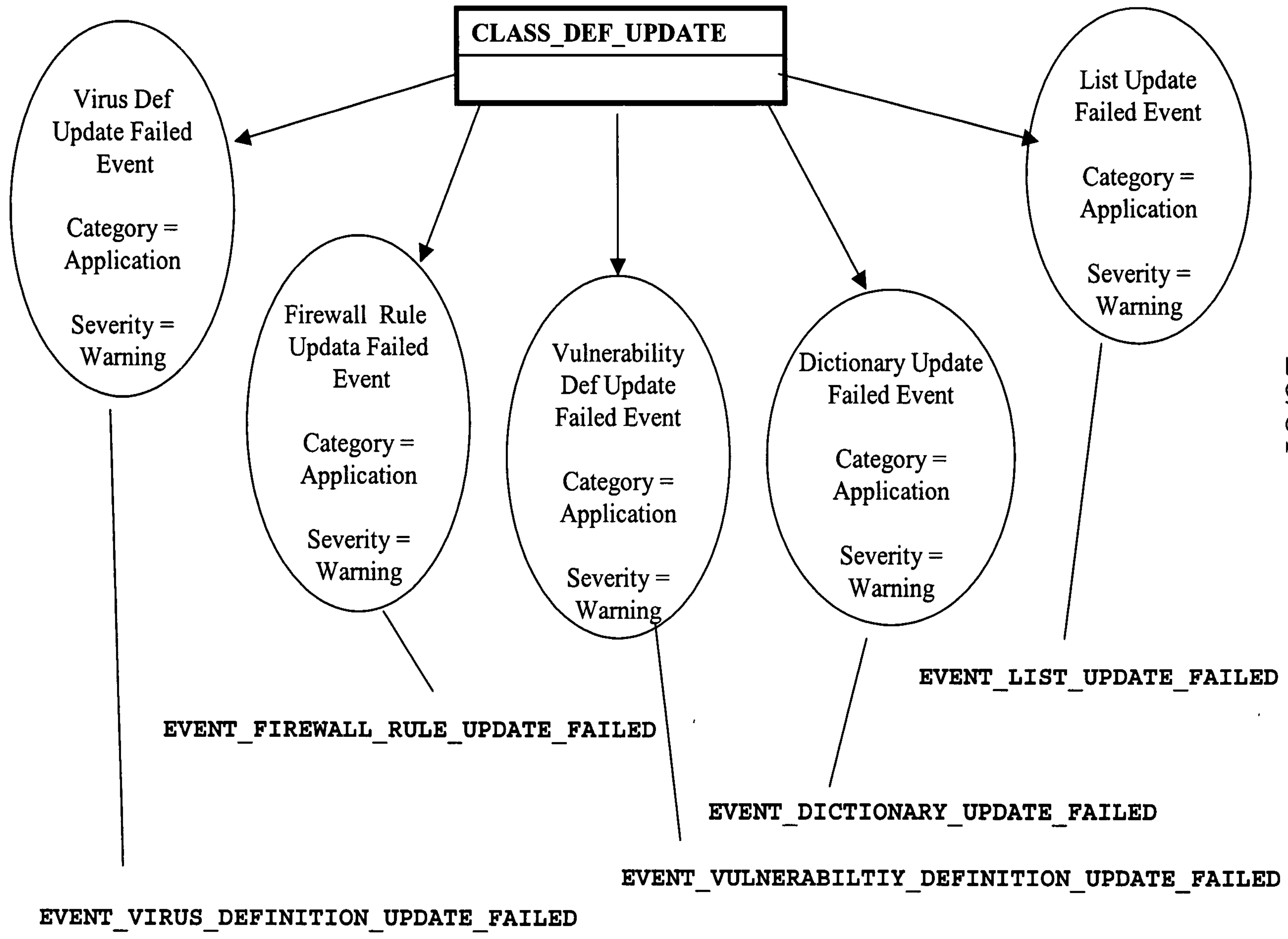

CLASS_DEF_UPDATE
Virus Def Update Failed Event
Category = Application
Severity = Warning
Firewall Rule Updata Failed Event
Category = Application
Severity = Warning
Vulnerability Def Update Failed Event
Category = Application
Severity = Warning
Dictionary Update Failed Event
Category = Application
Severity = Warning
List Update Failed Event
Category = Application
Severity = Warning
EVENT_LIST_UPDATE_FAILED
EVENT_FIREWALL_RULE_UPDATE_FAILED
EVENT_DICTIONARY_UPDATE_FAILED
EVENT_VULNERABILTIY_DEFINITION_UPDATE_FAILED
EVENT_VIRUS_DEFINITION_UPDATE_FAILED

FIG. 18B

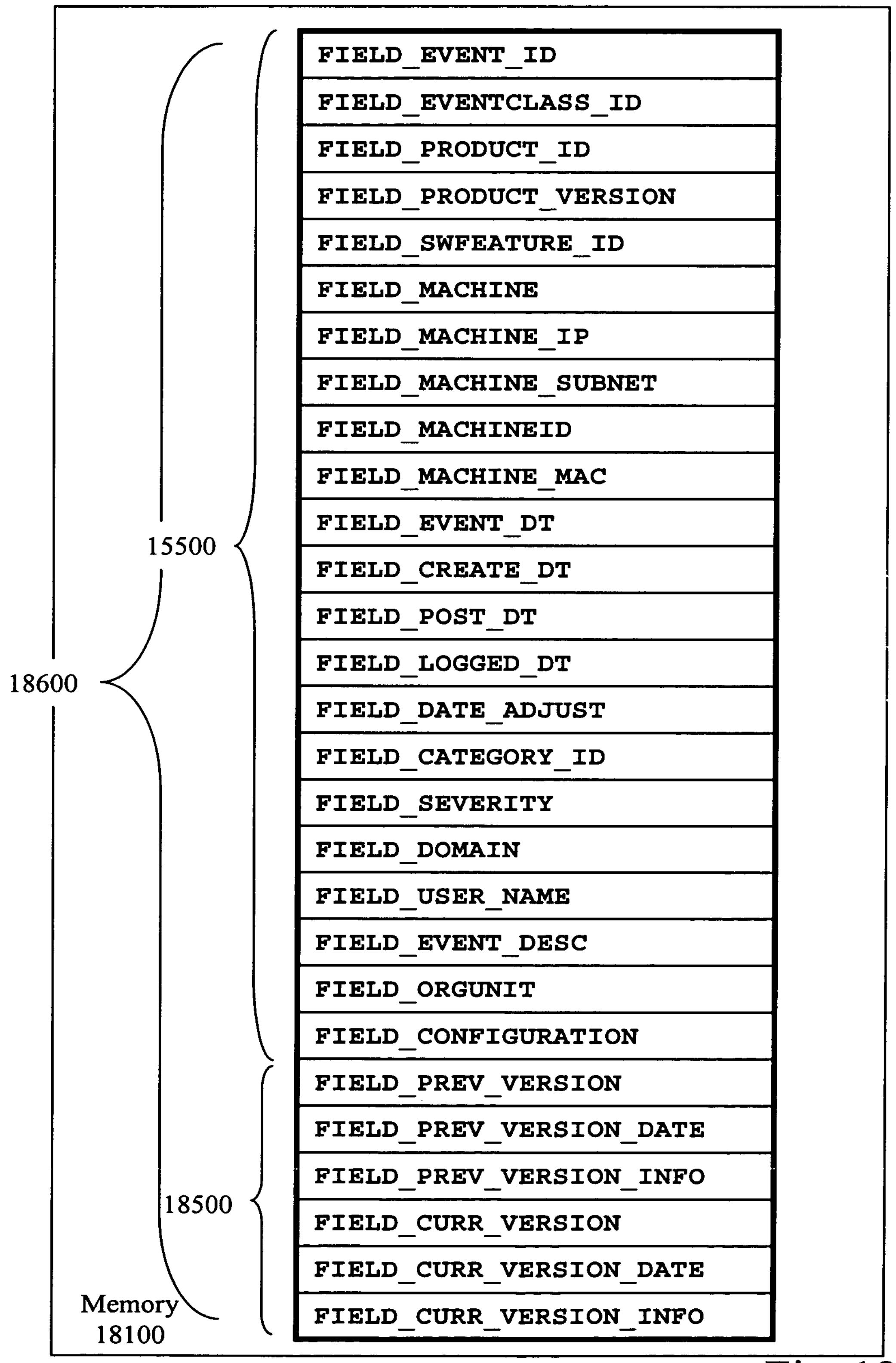

FIELD_EVENT_ID
FIELD_EVENTCLASS_ID
FIELD_PRODUCT_ID
FIELD_PRODUCT_VERSION
FIELD_SWFEATURE_ID
FIELD_MACHINE
FIELD_MACHINE_IP
FIELD_MACHINE_SUBNET
FIELD_MACHINEID
FIELD_MACHINE_MAC
FIELD_EVENT_DT
FIELD_CREATE_DT
FIELD_POST_DT
FIELD_LOGGED_DT
FIELD_DATE_ADJUST
FIELD_CATEGORY_ID
FIELD_SEVERITY
FIELD_DOMAIN
FIELD_USER_NAME
FIELD_EVENT_DESC
FIELD_ORGUNIT
FIELD_CONFIGURATION
FIELD_PREV_VERSION
FIELD_PREV_VERSION_DATE
FIELD_PREV_VERSION_INFO
FIELD_CURR_VERSION
FIELD_CURR_VERSION_DATE
FIELD_CURR_VERSION_INFO
15500
18600
18500
Memory
18100

Fig. 18C

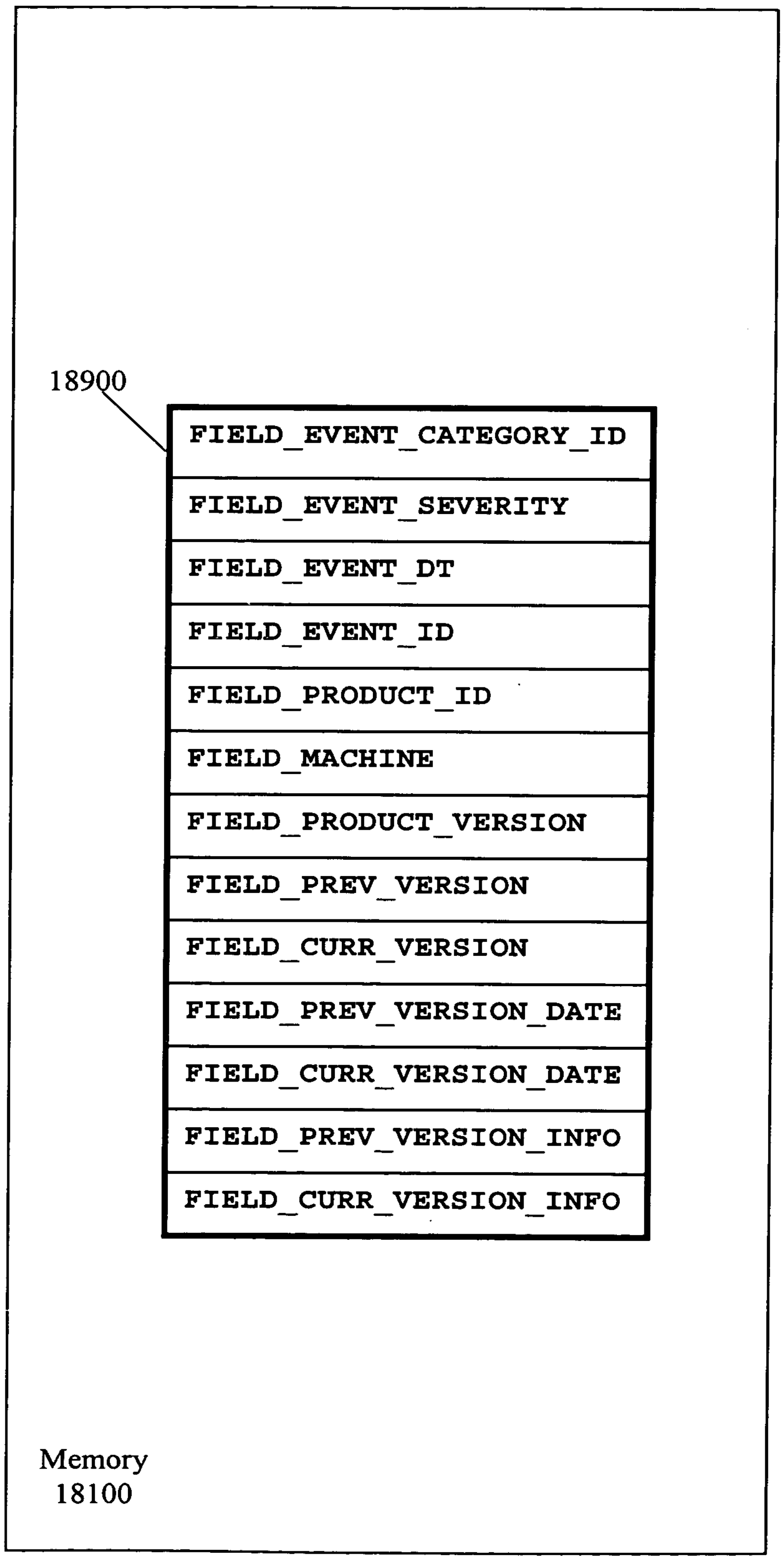

18900
FIELD_EVENT_CATEGORY_ID
FIELD_EVENT_SEVERITY
FIELD_EVENT_DT
FIELD_EVENT_ID
FIELD_PRODUCT_ID
FIELD_MACHINE
FIELD_PRODUCT_VERSION
FIELD_PREV_VERSION
FIELD_CURR_VERSION
FIELD_PREV_VERSION_DATE
FIELD_CURR_VERSION_DATE
FIELD_PREV_VERSION_INFO
FIELD_CURR_VERSION_INFO
Memory
18100

Fig. 18D

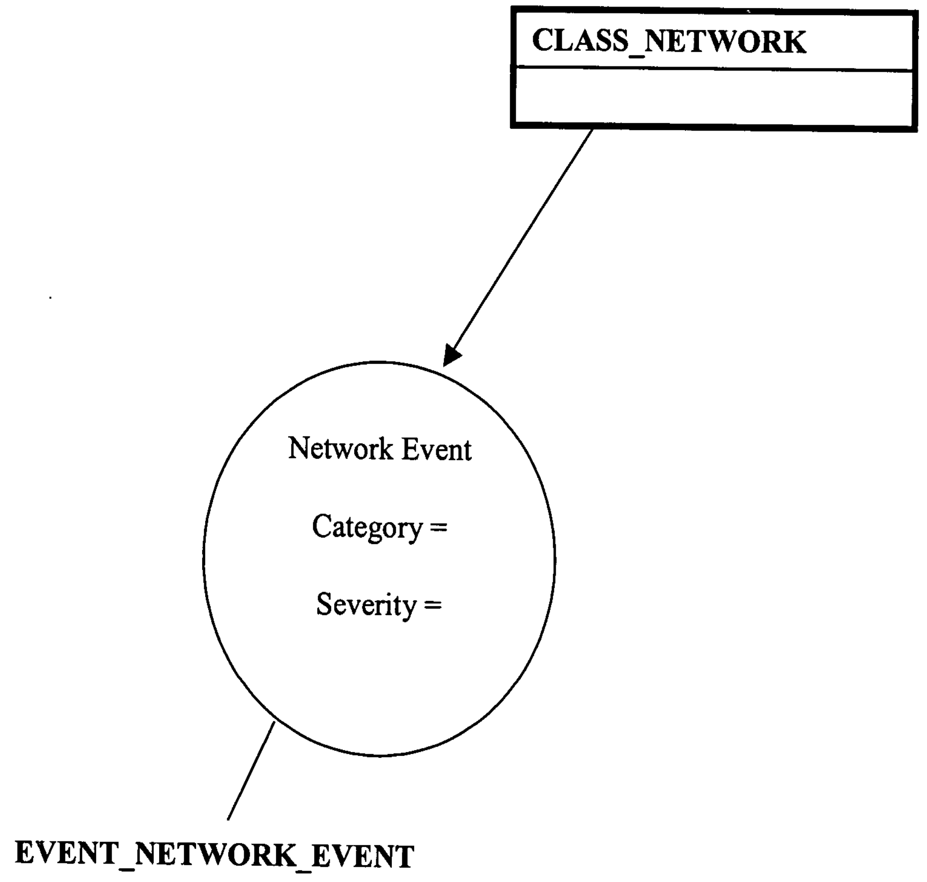

CLASS_NETWORK
Network Event
Category =
Severity =
EVENT_NETWORK_EVENT

FIG. 19A

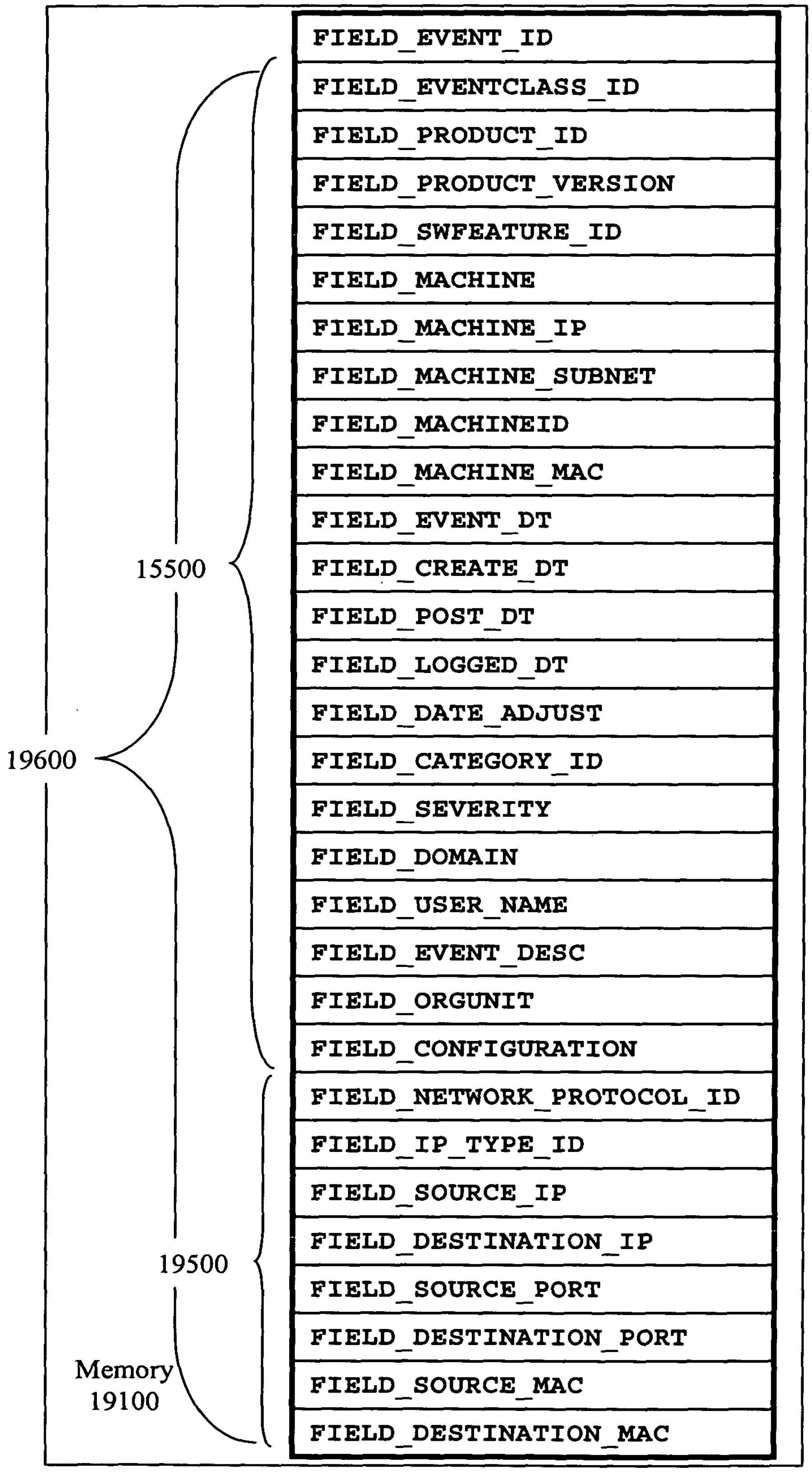

FIELD_EVENT_ID
FIELD_EVENTCLASS_ID
FIELD_PRODUCT_ID
FIELD_PRODUCT_VERSION
FIELD_SWFEATURE_ID
FIELD_MACHINE
FIELD_MACHINE_IP
FIELD_MACHINE_SUBNET
FIELD_MACHINEID
FIELD_MACHINE_MAC
FIELD_EVENT_DT
FIELD_CREATE_DT
FIELD_POST_DT
FIELD_LOGGED_DT
FIELD_DATE_ADJUST
FIELD_CATEGORY_ID
FIELD_SEVERITY
FIELD_DOMAIN
FIELD_USER_NAME
FIELD_EVENT_DESC
FIELD_ORGUNIT
FIELD_CONFIGURATION
FIELD_NETWORK_PROTOCOL_ID
FIELD_IP_TYPE_ID
FIELD_SOURCE_IP
FIELD_DESTINATION_IP
FIELD_SOURCE_PORT
FIELD_DESTINATION_PORT
FIELD_SOURCE_MAC
FIELD_DESTINATION_MAC
15500
19600
19500
Memory
19100

Fig. 19B

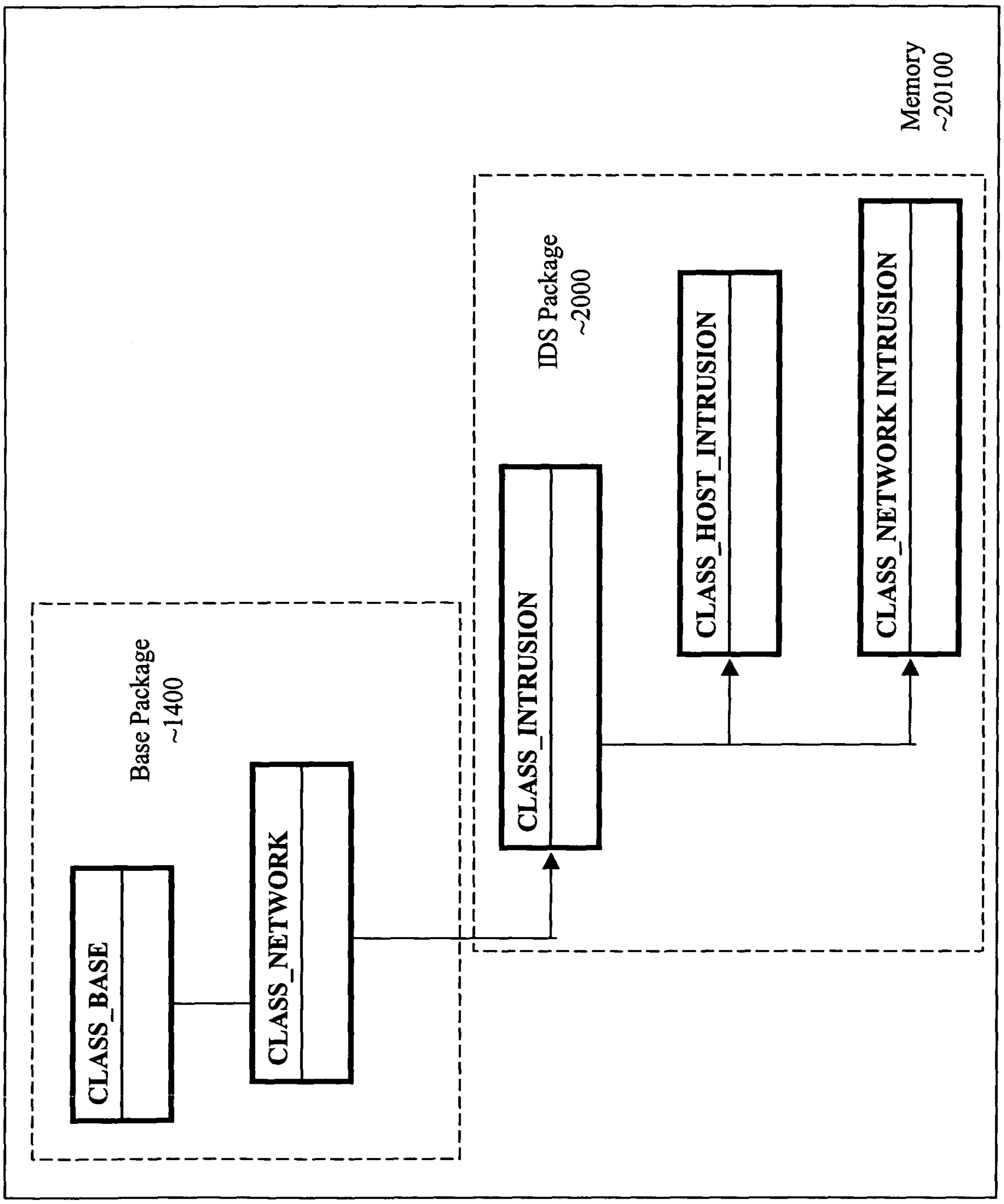

Memory
~20100
IDS Package
~2000
Base Package
~1400
CLASS_BASE
CLASS_NETWORK
CLASS_INTRUSION
CLASS_HOST_INTRUSION
CLASS_NETWORK INTRUSION

FIG. 20

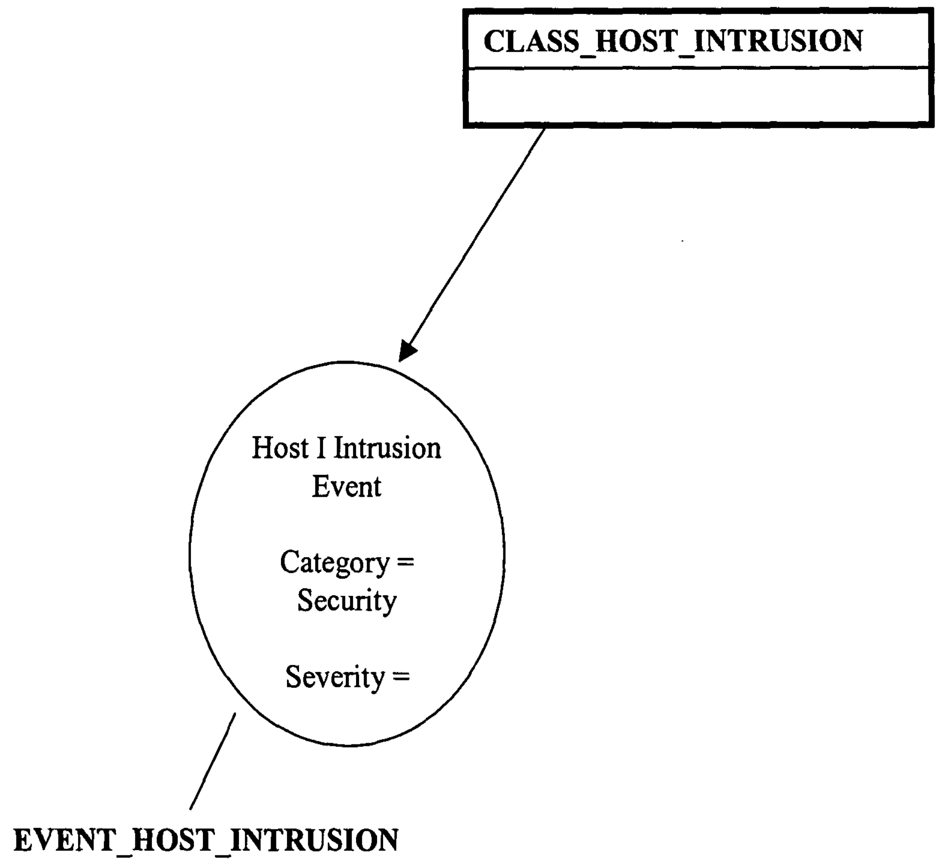

CLASS_HOST_INTRUSION
Host I Intrusion Event
Category = Security
Severity =
EVENT_HOST_INTRUSION

FIG. 21A

Fig. 21B_1

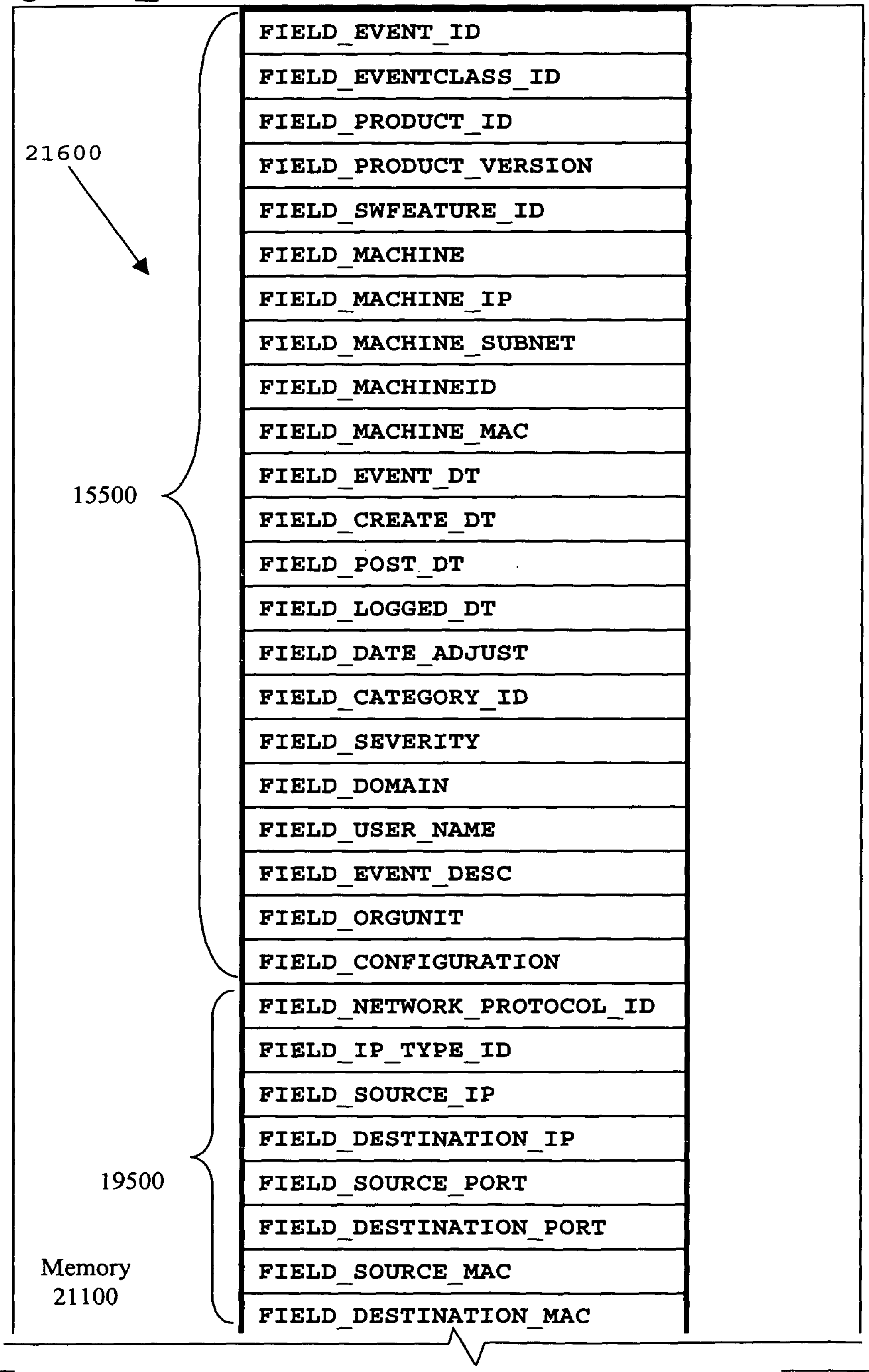
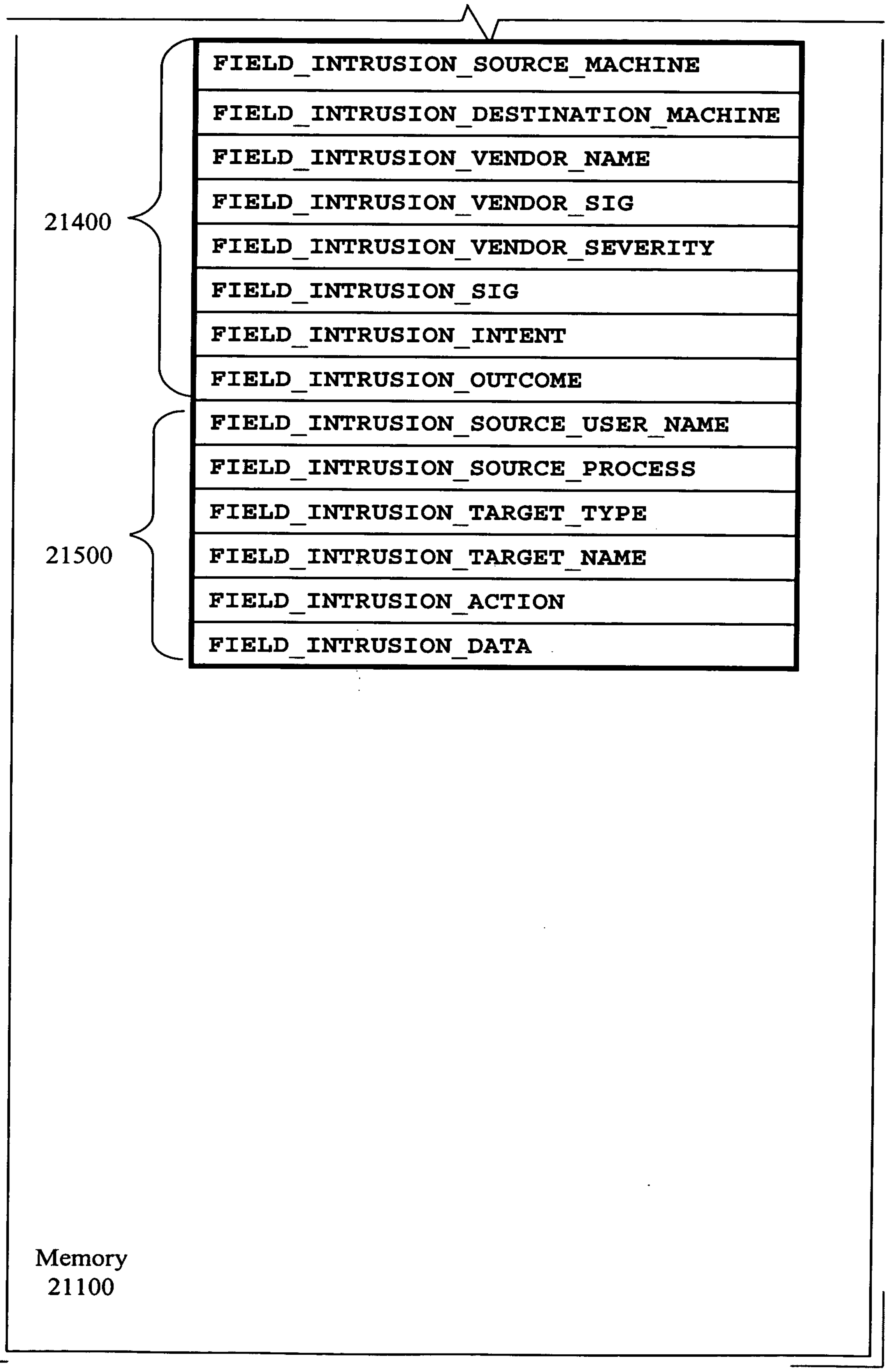

21600
15500
19500
Memory
21100
FIELD_EVENT_ID
FIELD_EVENTCLASS_ID
FIELD_PRODUCT_ID
FIELD_PRODUCT_VERSION
FIELD_SWFEATURE_ID
FIELD_MACHINE
FIELD_MACHINE_IP
FIELD_MACHINE_SUBNET
FIELD_MACHINEID
FIELD_MACHINE_MAC
FIELD_EVENT_DT
FIELD_CREATE_DT
FIELD_POST_DT
FIELD_LOGGED_DT
FIELD_DATE_ADJUST
FIELD_CATEGORY_ID
FIELD_SEVERITY
FIELD_DOMAIN
FIELD_USER_NAME
FIELD_EVENT_DESC
FIELD_ORGUNIT
FIELD_CONFIGURATION
FIELD_NETWORK_PROTOCOL_ID
FIELD_IP_TYPE_ID
FIELD_SOURCE_IP
FIELD_DESTINATION_IP
FIELD_SOURCE_PORT
FIELD_DESTINATION_PORT
FIELD_SOURCE_MAC
FIELD_DESTINATION_MAC

Fig. 21B_2



FIELD_INTRUSION_SOURCE_MACHINE
FIELD_INTRUSION_DESTINATION_MACHINE
FIELD_INTRUSION_VENDOR_NAME
FIELD_INTRUSION_VENDOR_SIG
FIELD_INTRUSION_VENDOR_SEVERITY
FIELD_INTRUSION_SIG
FIELD_INTRUSION_INTENT
FIELD_INTRUSION_OUTCOME
FIELD_INTRUSION_SOURCE_USER_NAME
FIELD_INTRUSION_SOURCE_PROCESS
FIELD_INTRUSION_TARGET_TYPE
FIELD_INTRUSION_TARGET_NAME
FIELD_INTRUSION_ACTION
FIELD_INTRUSION_DATA
21400
21500
Memory
21100

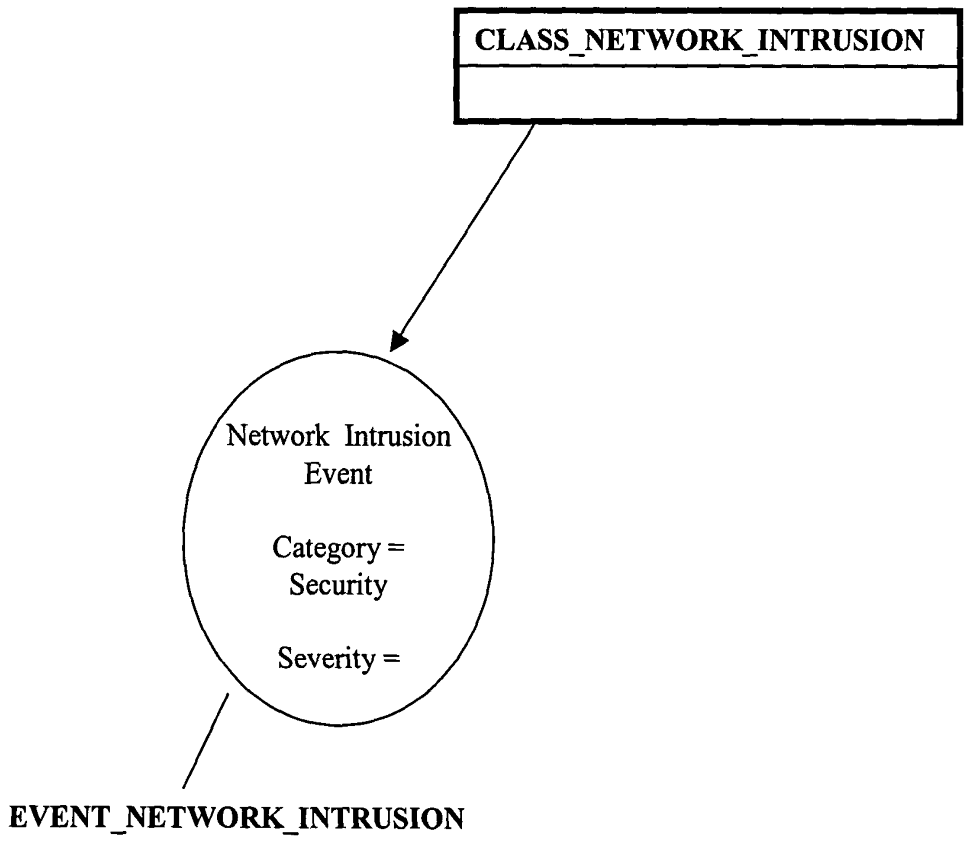
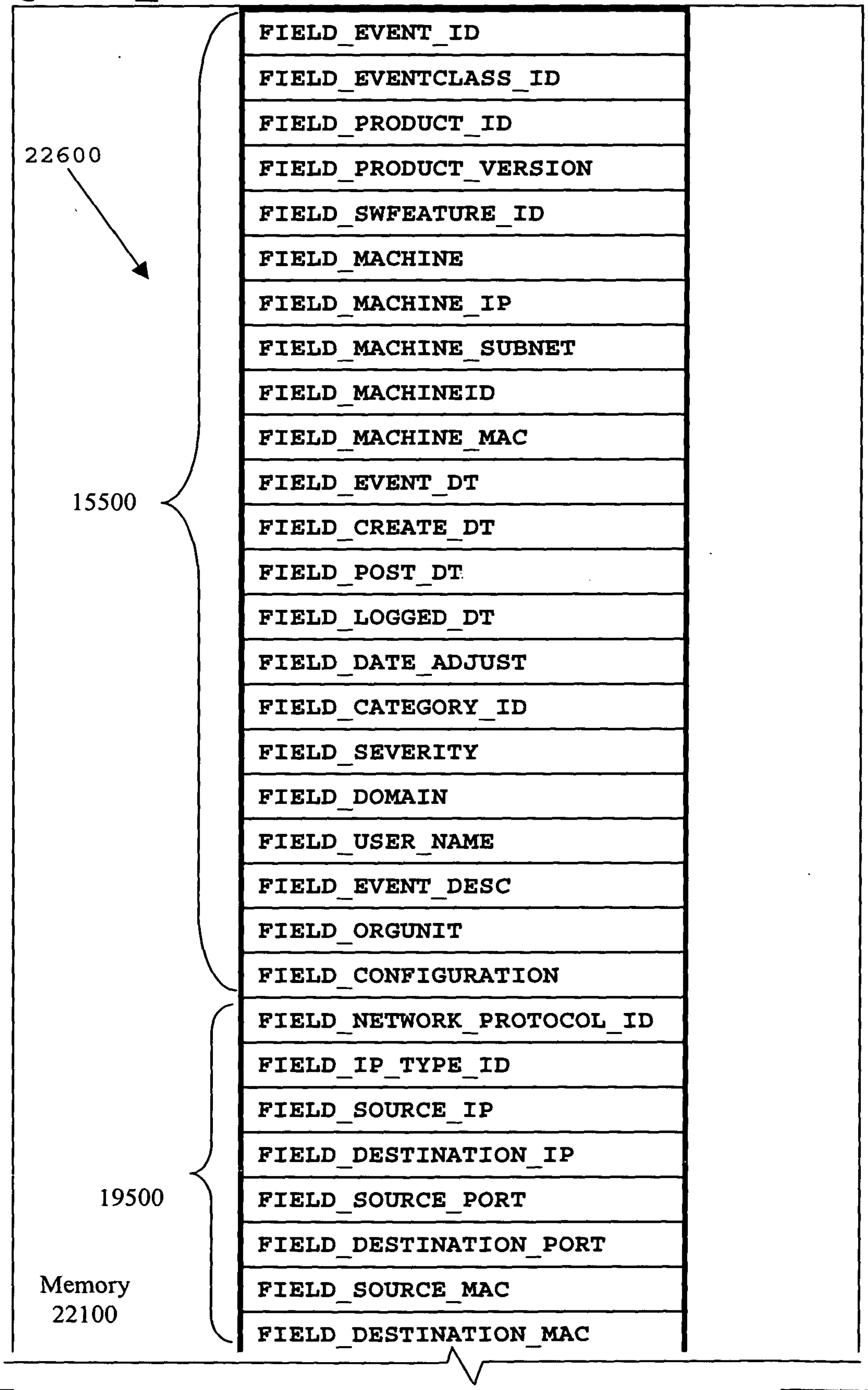

CLASS_NETWORK_INTRUSION
Network Intrusion Event
Category = Security
Severity =
EVENT_NETWORK_INTRUSION

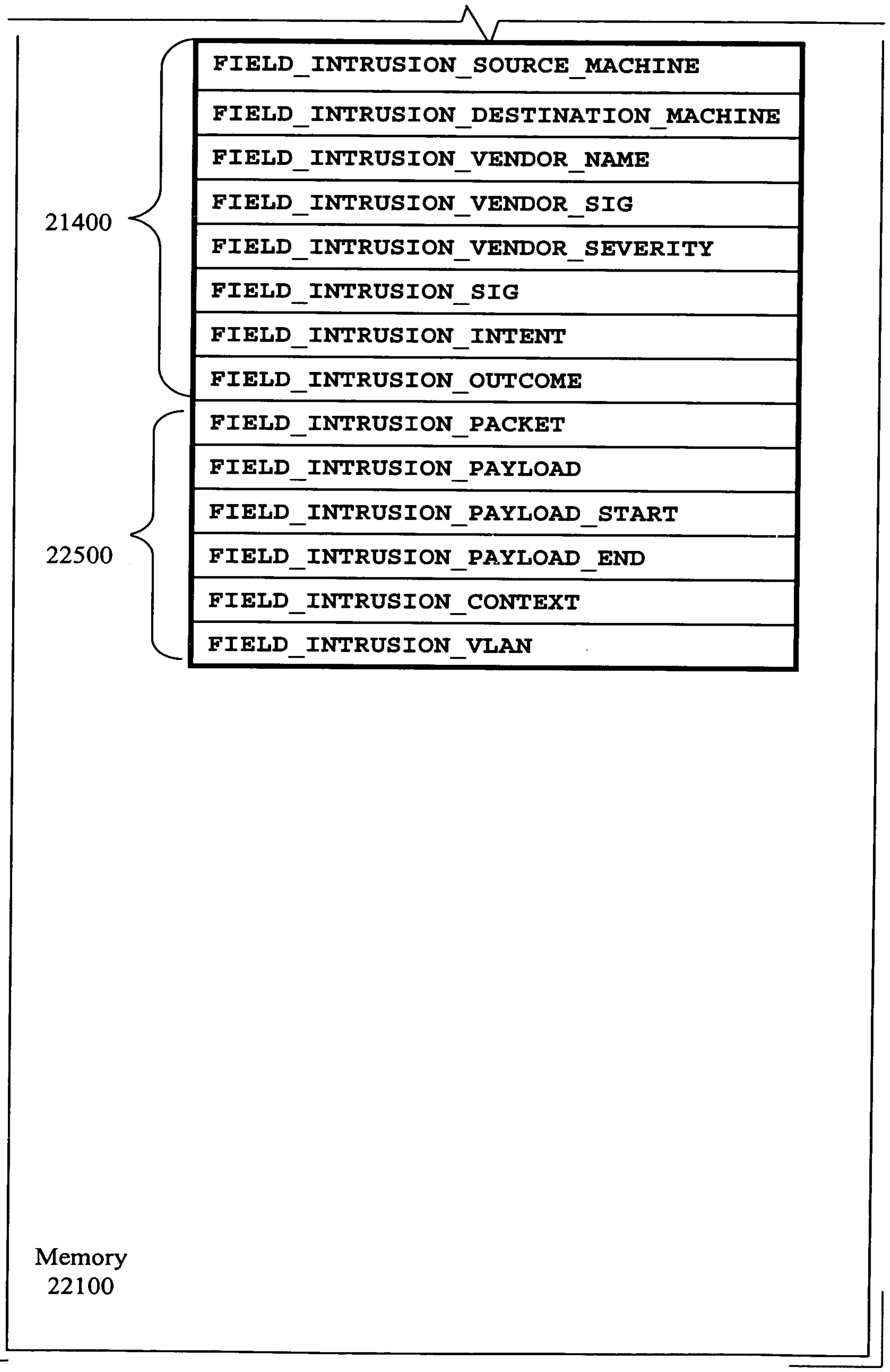
FIG. 22A

Fig. 22B_1



22600
15500
19500
Memory
22100
FIELD_EVENT_ID
FIELD_EVENTCLASS_ID
FIELD_PRODUCT_ID
FIELD_PRODUCT_VERSION
FIELD_SWFEATURE_ID
FIELD_MACHINE
FIELD_MACHINE_IP
FIELD_MACHINE_SUBNET
FIELD_MACHINEID
FIELD_MACHINE_MAC
FIELD_EVENT_DT
FIELD_CREATE_DT
FIELD_POST_DT
FIELD_LOGGED_DT
FIELD_DATE_ADJUST
FIELD_CATEGORY_ID
FIELD_SEVERITY
FIELD_DOMAIN
FIELD_USER_NAME
FIELD_EVENT_DESC
FIELD_ORGUNIT
FIELD_CONFIGURATION
FIELD_NETWORK_PROTOCOL_ID
FIELD_IP_TYPE_ID
FIELD_SOURCE_IP
FIELD_DESTINATION_IP
FIELD_SOURCE_PORT
FIELD_DESTINATION_PORT
FIELD_SOURCE_MAC
FIELD_DESTINATION_MAC

Fig. 22B_2



FIELD_INTRUSION_SOURCE_MACHINE
FIELD_INTRUSION_DESTINATION_MACHINE
FIELD_INTRUSION_VENDOR_NAME
FIELD_INTRUSION_VENDOR_SIG
FIELD_INTRUSION_VENDOR_SEVERITY
FIELD_INTRUSION_SIG
FIELD_INTRUSION_INTENT
FIELD_INTRUSION_OUTCOME
FIELD_INTRUSION_PACKET
FIELD_INTRUSION_PAYLOAD
FIELD_INTRUSION_PAYLOAD_START
FIELD_INTRUSION_PAYLOAD_END
FIELD_INTRUSION_CONTEXT
FIELD_INTRUSION_VLAN
21400
22500
Memory
22100

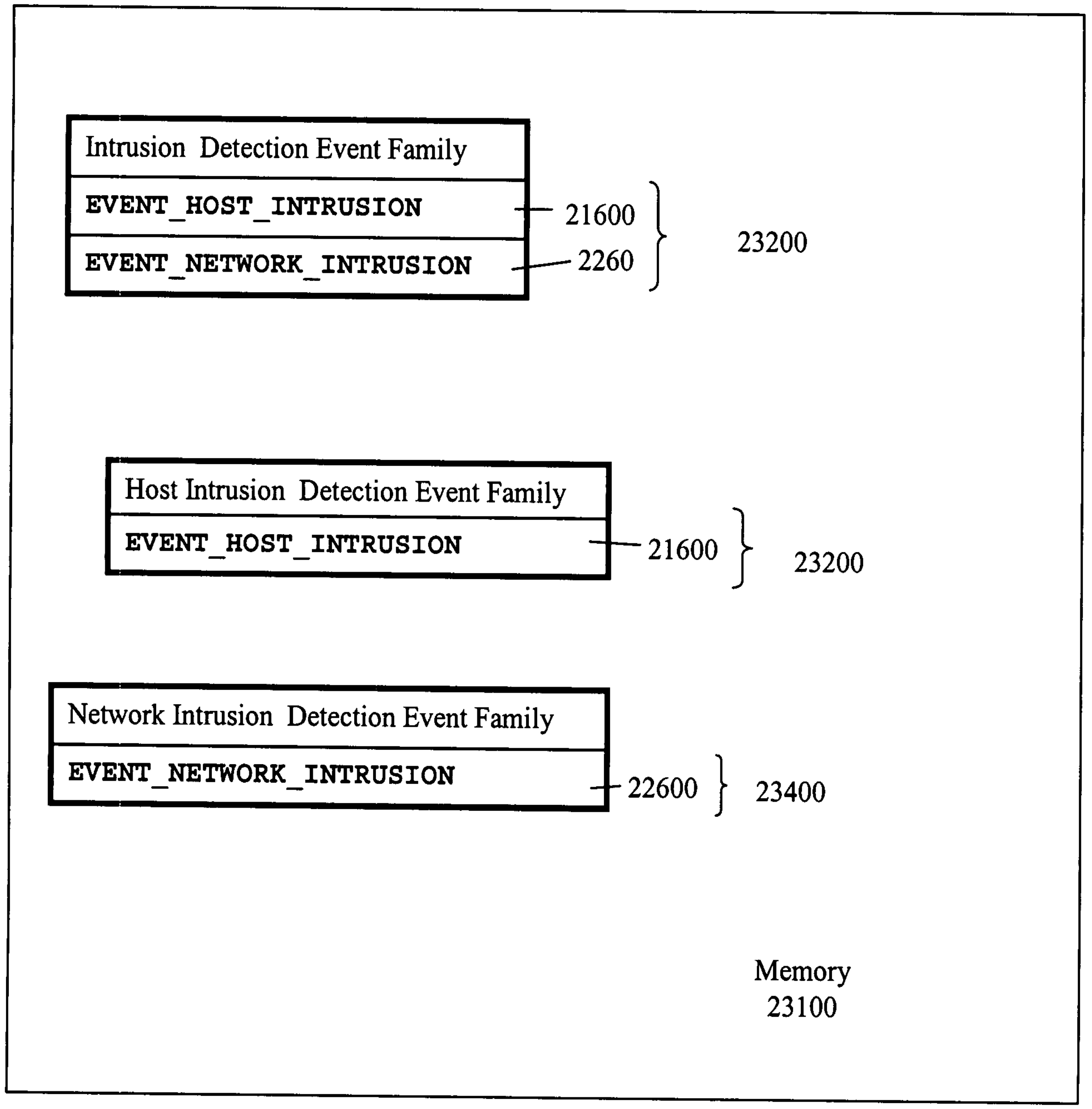
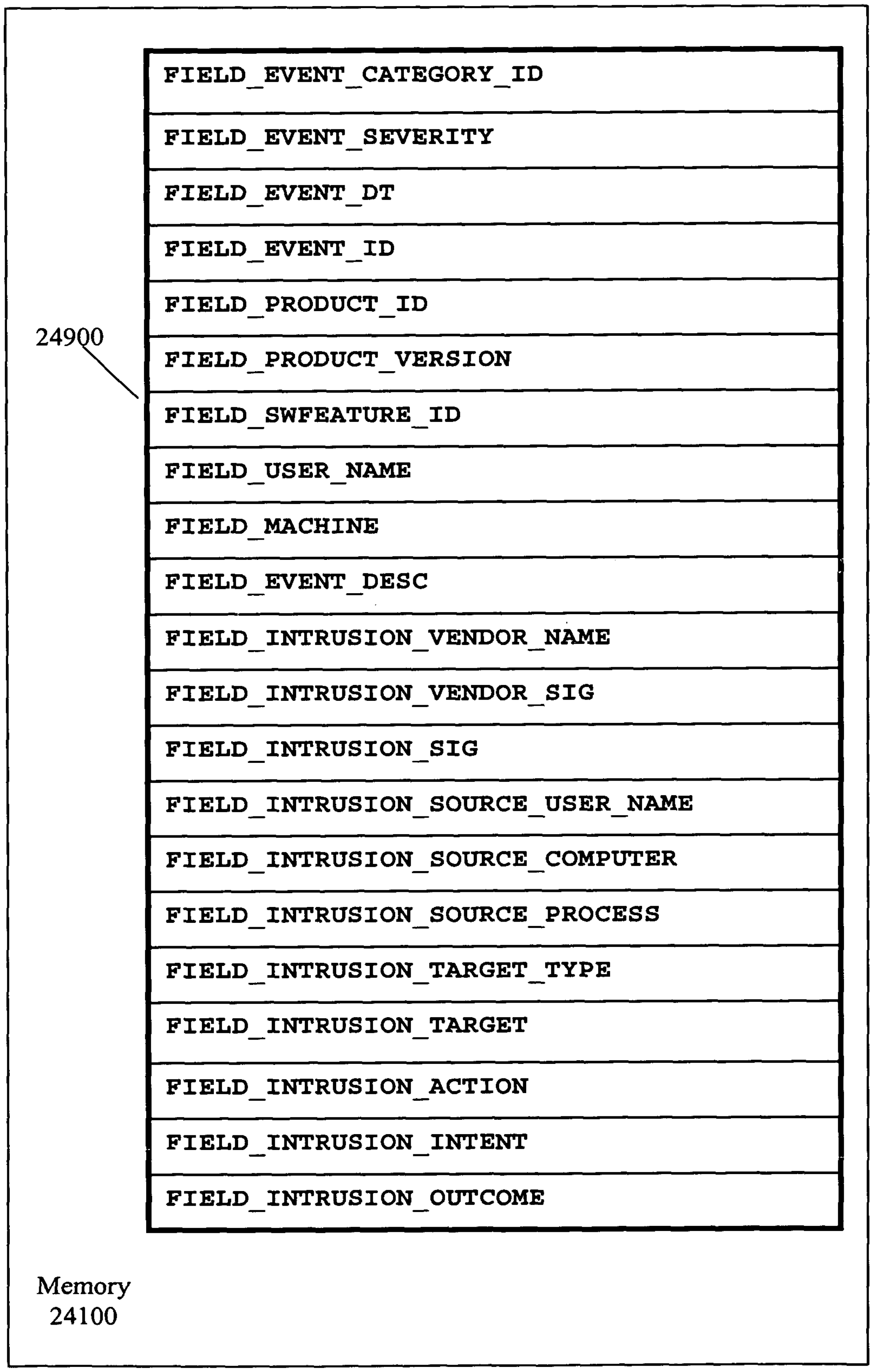

Intrusion Detection Event Family
EVENT_HOST_INTRUSION
EVENT_NETWORK_INTRUSION
21600
2260
23200
Host Intrusion Detection Event Family
EVENT_HOST_INTRUSION
21600
23200
Network Intrusion Detection Event Family
EVENT_NETWORK_INTRUSION
22600
23400
Memory
23100

FIG. 23

Fig. 24



FIELD_EVENT_CATEGORY_ID
FIELD_EVENT_SEVERITY
FIELD_EVENT_DT
FIELD_EVENT_ID
FIELD_PRODUCT_ID
FIELD_PRODUCT_VERSION
FIELD_SWFEATURE_ID
FIELD_USER_NAME
FIELD_MACHINE
FIELD_EVENT_DESC
FIELD_INTRUSION_VENDOR_NAME
FIELD_INTRUSION_VENDOR_SIG
FIELD_INTRUSION_SIG
FIELD_INTRUSION_SOURCE_USER_NAME
FIELD_INTRUSION_SOURCE_COMPUTER
FIELD_INTRUSION_SOURCE_PROCESS
FIELD_INTRUSION_TARGET_TYPE
FIELD_INTRUSION_TARGET
FIELD_INTRUSION_ACTION
FIELD_INTRUSION_INTENT
FIELD_INTRUSION_OUTCOME

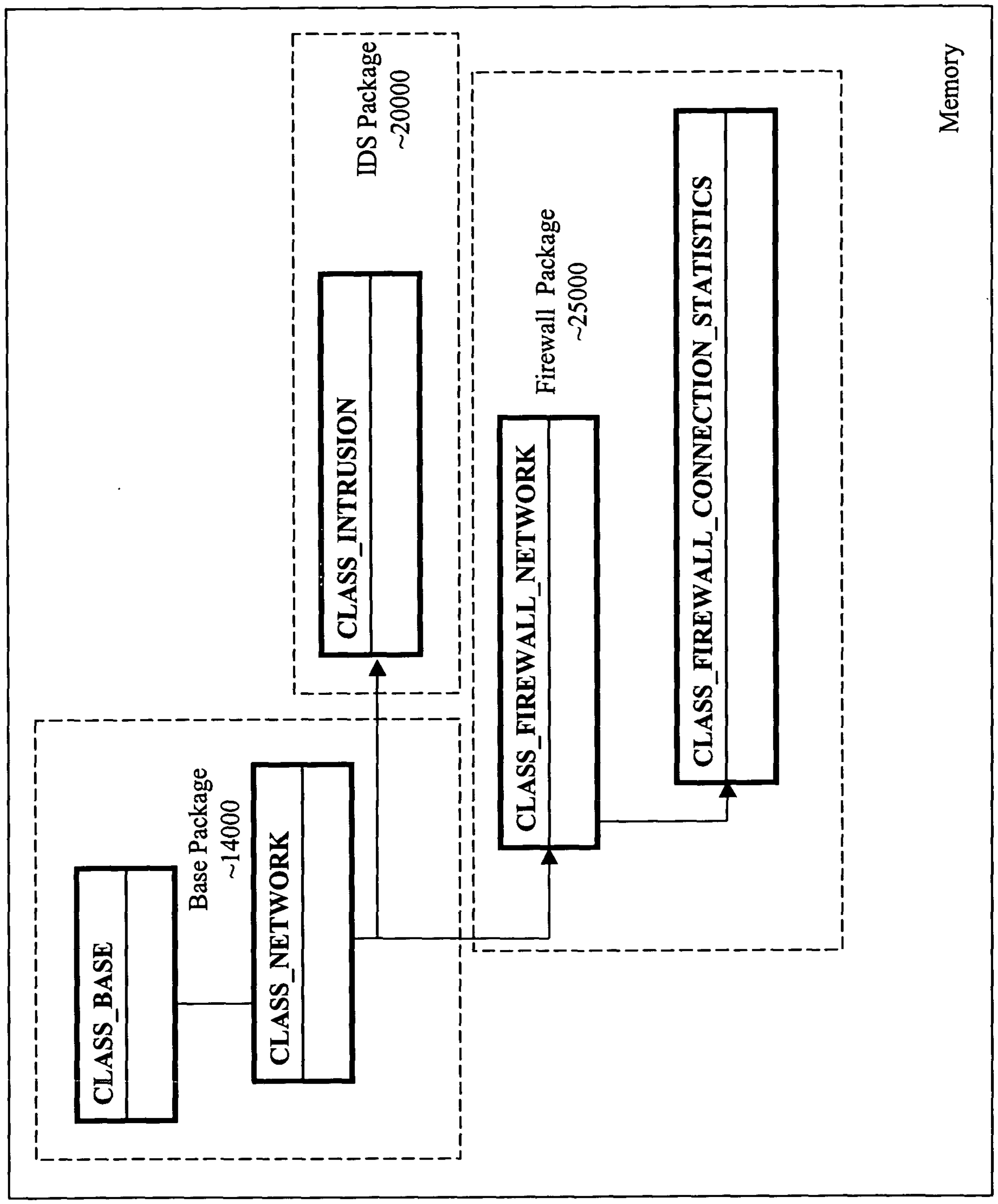

Memory
IDS Package
~20000
CLASS_INTRUSION
Firewall Package
~25000
CLASS_FIREWALL_NETWORK
CLASS_FIREWALL_CONNECTION_STATISTICS
Base Package
~14000
CLASS_BASE
CLASS_NETWORK

FIG. 25

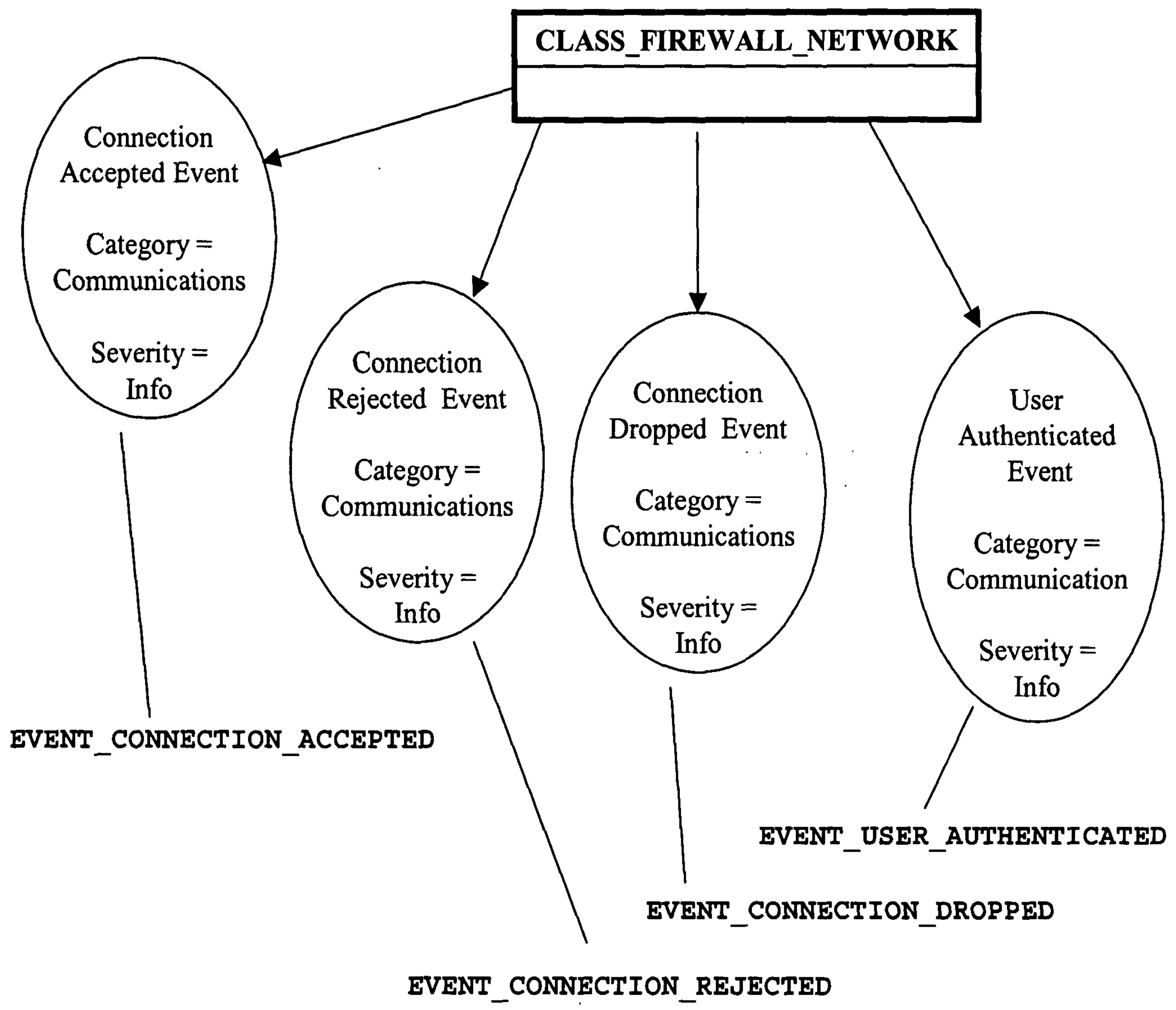

CLASS_FIREWALL_NETWORK
Connection Accepted Event
Category = Communications
Severity = Info
Connection Rejected Event
Category = Communications
Severity = Info
Connection Dropped Event
Category = Communications
Severity = Info
User Authenticated Event
Category = Communication
Severity = Info
EVENT_CONNECTION_ACCEPTED
EVENT_CONNECTION_REJECTED
EVENT_CONNECTION_DROPPED
EVENT_USER_AUTHENTICATED

FIG. 26A

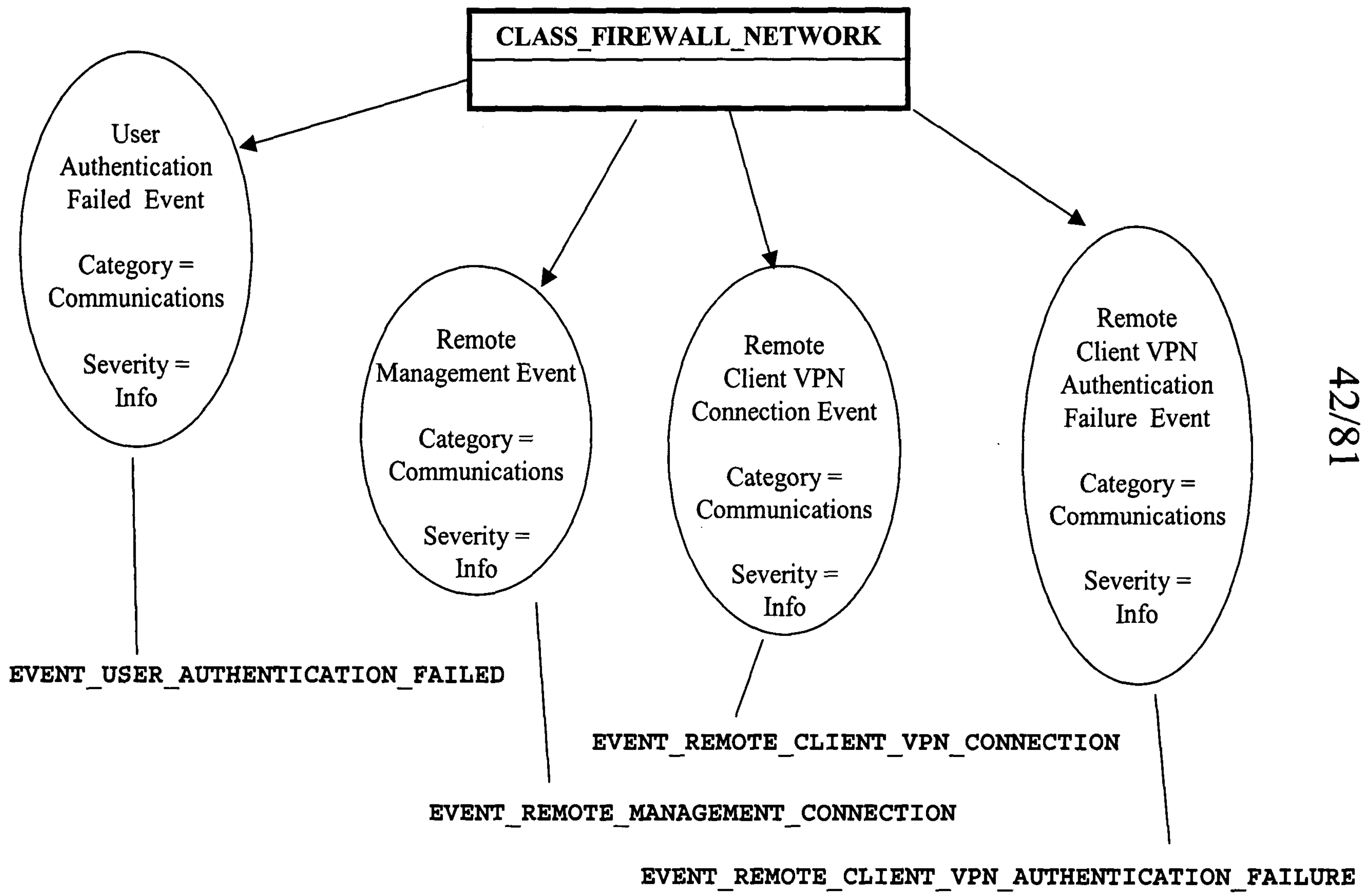
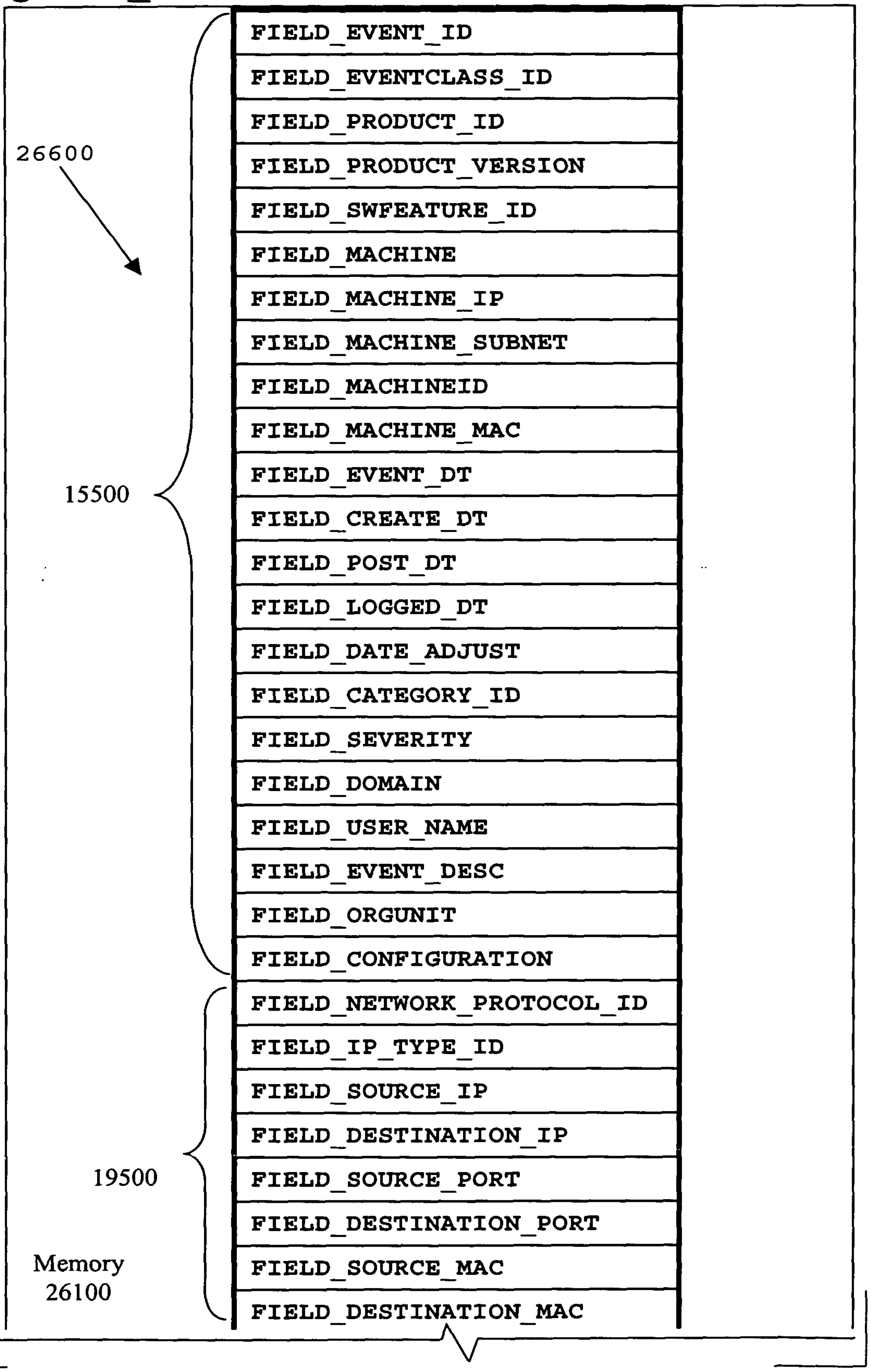

CLASS_FIREWALL_NETWORK
User Authentication Failed Event
Category = Communications
Severity = Info
Remote Management Event
Category = Communications
Severity = Info
Remote Client VPN Connection Event
Category = Communications
Severity = Info
Remote Client VPN Authentication Failure Event
Category = Communications
Severity = Info
EVENT_USER_AUTHENTICATION_FAILED
EVENT_REMOTE_CLIENT_VPN_CONNECTION
EVENT_REMOTE_MANAGEMENT_CONNECTION
EVENT_REMOTE_CLIENT_VPN_AUTHENTICATION_FAILURE

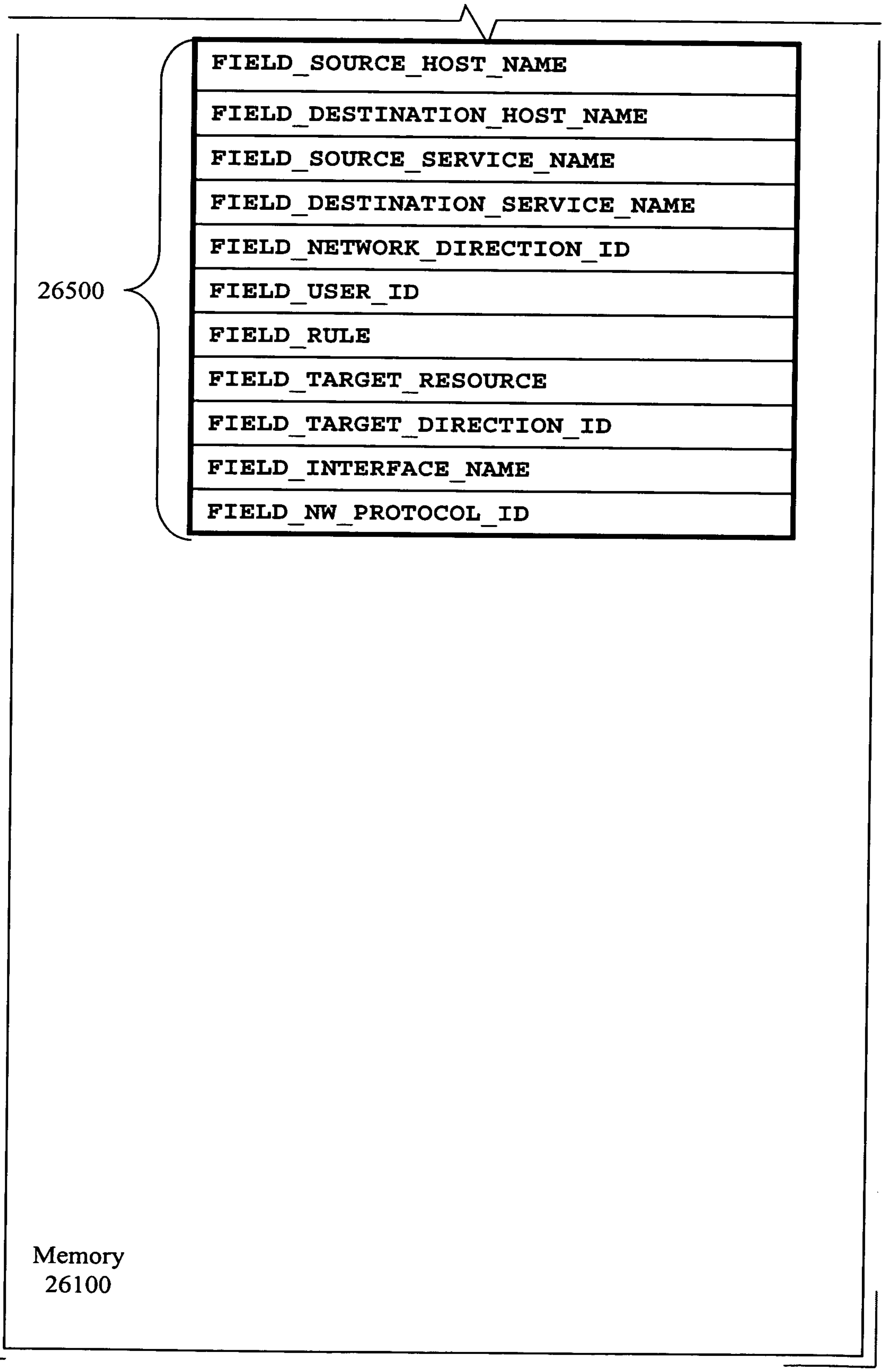
FIG. 26B

Fig. 26C_1



26600
15500
19500
Memory 26100
FIELD_EVENT_ID
FIELD_EVENTCLASS_ID
FIELD_PRODUCT_ID
FIELD_PRODUCT_VERSION
FIELD_SWFEATURE_ID
FIELD_MACHINE
FIELD_MACHINE_IP
FIELD_MACHINE_SUBNET
FIELD_MACHINEID
FIELD_MACHINE_MAC
FIELD_EVENT_DT
FIELD_CREATE_DT
FIELD_POST_DT
FIELD_LOGGED_DT
FIELD_DATE_ADJUST
FIELD_CATEGORY_ID
FIELD_SEVERITY
FIELD_DOMAIN
FIELD_USER_NAME
FIELD_EVENT_DESC
FIELD_ORGUNIT
FIELD_CONFIGURATION
FIELD_NETWORK_PROTOCOL_ID
FIELD_IP_TYPE_ID
FIELD_SOURCE_IP
FIELD_DESTINATION_IP
FIELD_SOURCE_PORT
FIELD_DESTINATION_PORT
FIELD_SOURCE_MAC
FIELD_DESTINATION_MAC

Fig. 26C_2


FIELD_SOURCE_HOST_NAME
FIELD_DESTINATION_HOST_NAME
FIELD_SOURCE_SERVICE_NAME
FIELD_DESTINATION_SERVICE_NAME
FIELD_NETWORK_DIRECTION_ID
FIELD_USER_ID
FIELD_RULE
FIELD_TARGET_RESOURCE
FIELD_TARGET_DIRECTION_ID
FIELD_INTERFACE_NAME
FIELD_NW_PROTOCOL_ID
26500
Memory
26100

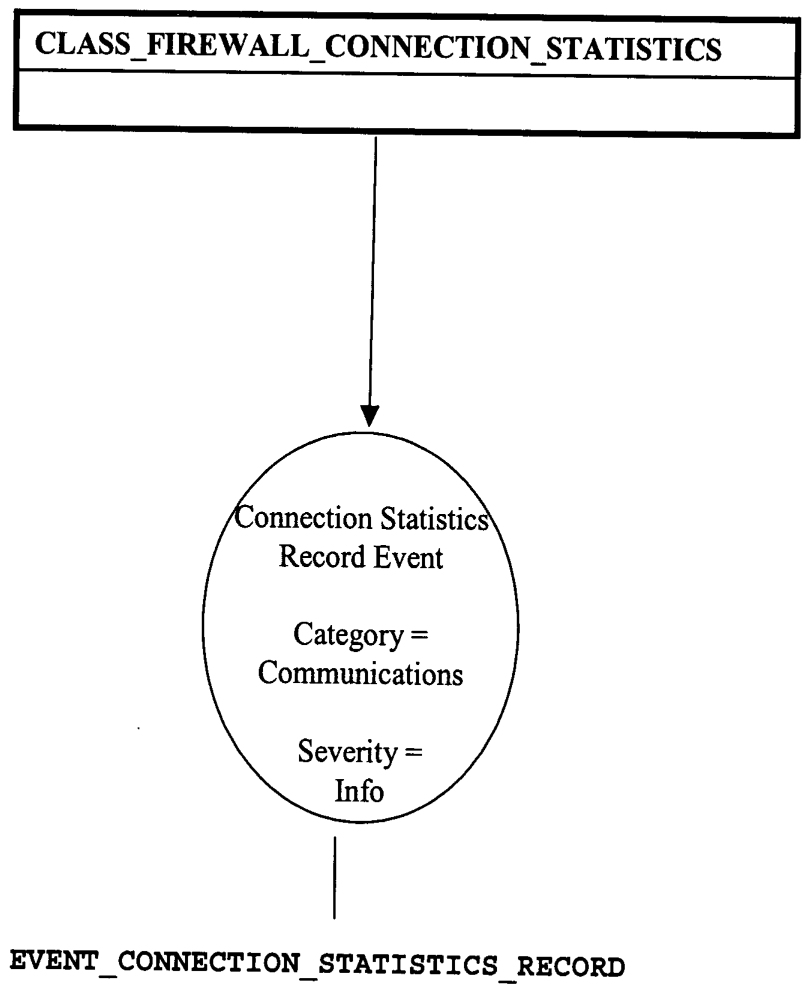

CLASS_FIREWALL_CONNECTION_STATISTICS
Connection Statistics Record Event
Category = Communications
Severity = Info
EVENT_CONNECTION_STATISTICS_RECORD

FIG. 27A

Fig. 27B_1

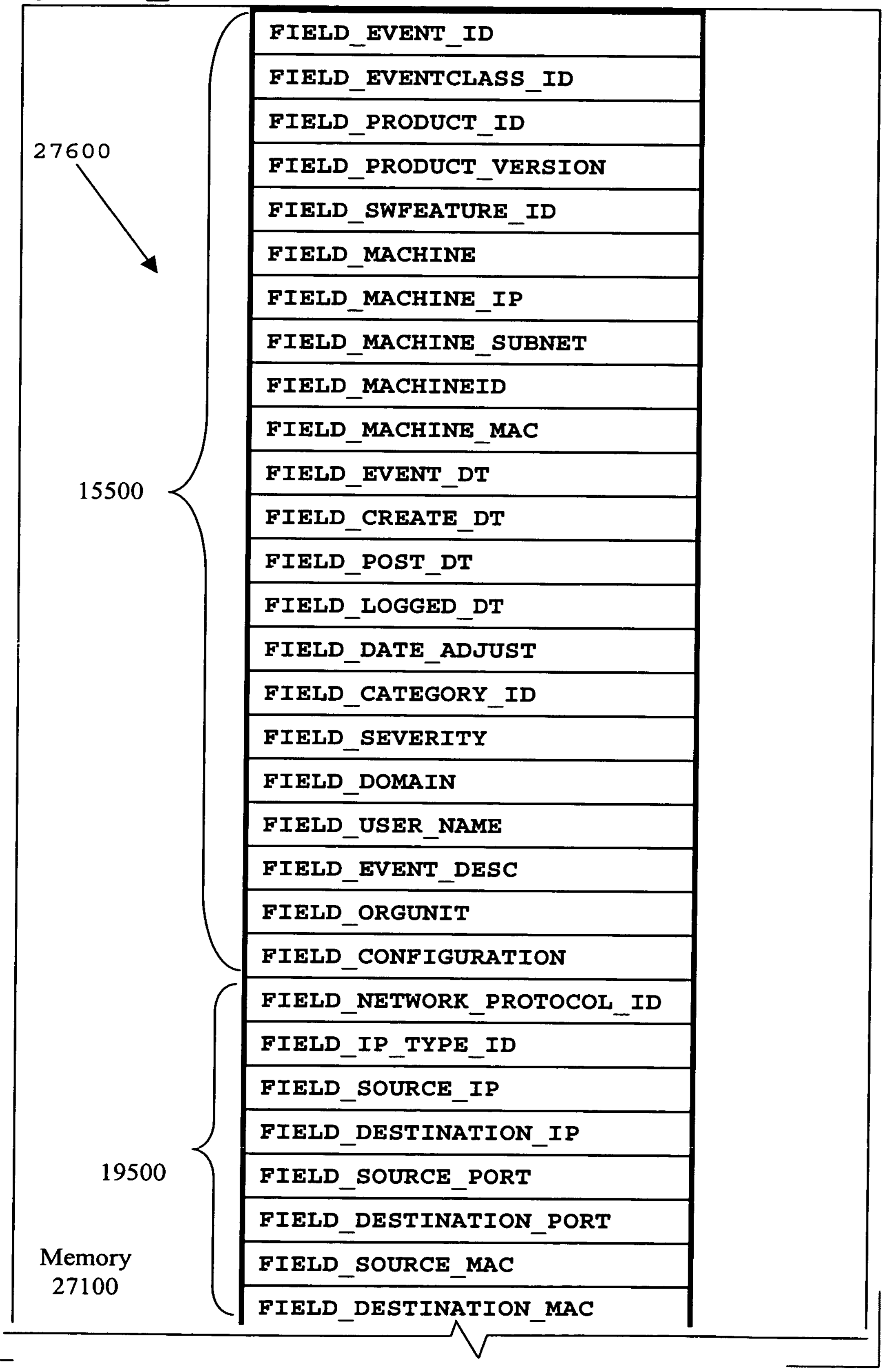

27600
15500
19500
Memory
27100
FIELD_EVENT_ID
FIELD_EVENTCLASS_ID
FIELD_PRODUCT_ID
FIELD_PRODUCT_VERSION
FIELD_SWFEATURE_ID
FIELD_MACHINE
FIELD_MACHINE_IP
FIELD_MACHINE_SUBNET
FIELD_MACHINEID
FIELD_MACHINE_MAC
FIELD_EVENT_DT
FIELD_CREATE_DT
FIELD_POST_DT
FIELD_LOGGED_DT
FIELD_DATE_ADJUST
FIELD_CATEGORY_ID
FIELD_SEVERITY
FIELD_DOMAIN
FIELD_USER_NAME
FIELD_EVENT_DESC
FIELD_ORGUNIT
FIELD_CONFIGURATION
FIELD_NETWORK_PROTOCOL_ID
FIELD_IP_TYPE_ID
FIELD_SOURCE_IP
FIELD_DESTINATION_IP
FIELD_SOURCE_PORT
FIELD_DESTINATION_PORT
FIELD_SOURCE_MAC
FIELD_DESTINATION_MAC

Fig. 27B_2

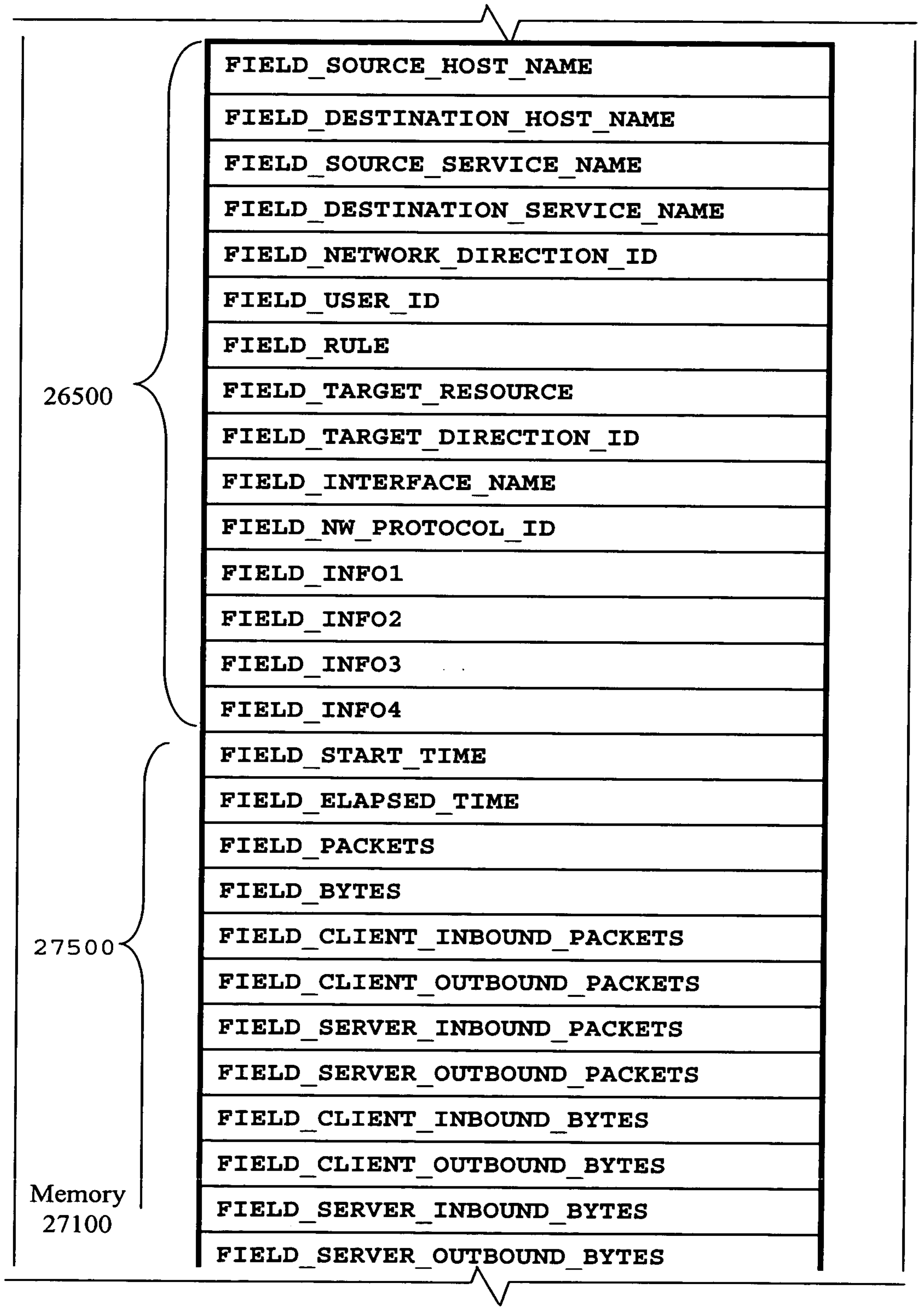
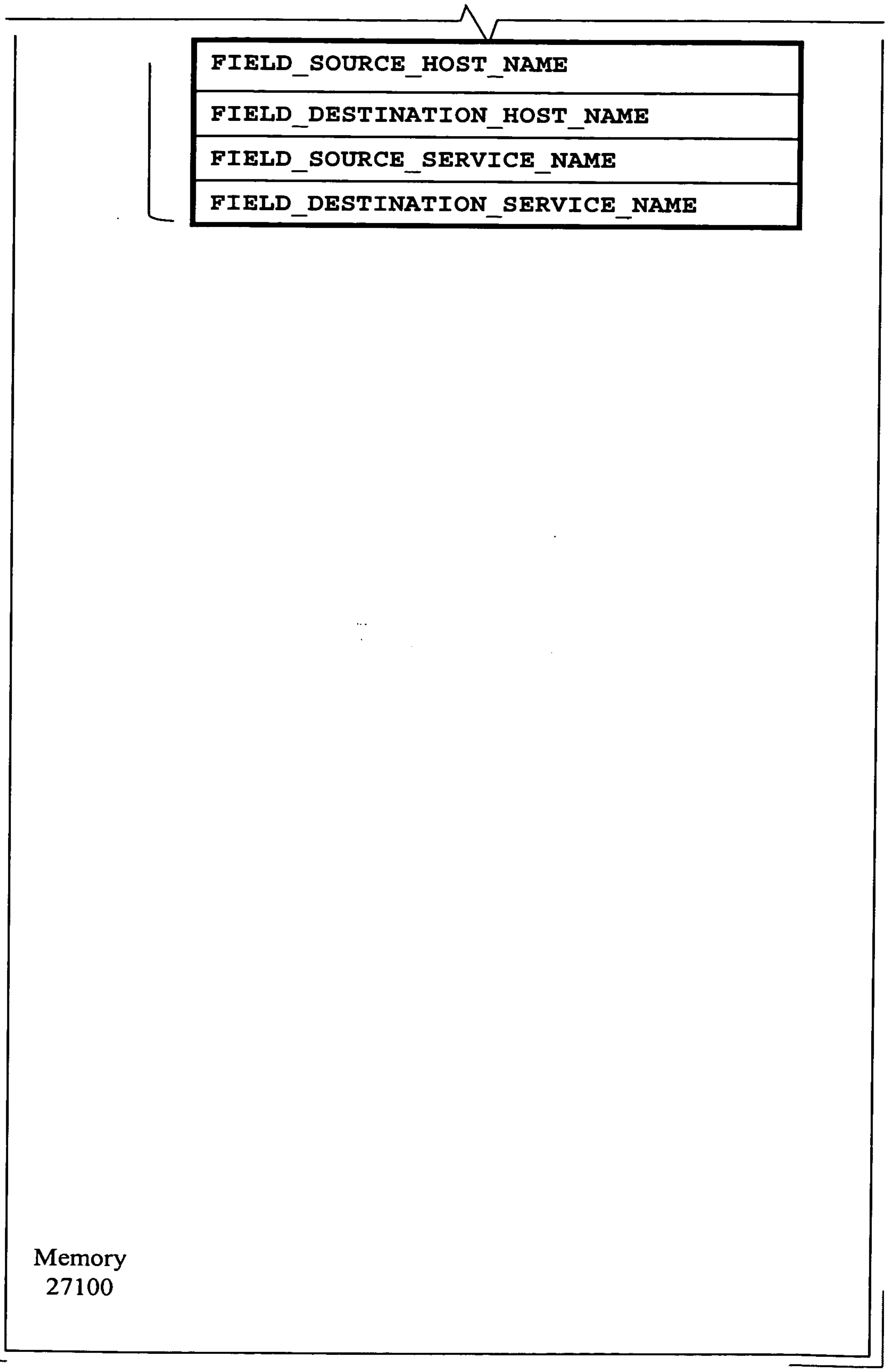

FIELD_SOURCE_HOST_NAME
FIELD_DESTINATION_HOST_NAME
FIELD_SOURCE_SERVICE_NAME
FIELD_DESTINATION_SERVICE_NAME
FIELD_NETWORK_DIRECTION_ID
FIELD_USER_ID
FIELD_RULE
FIELD_TARGET_RESOURCE
FIELD_TARGET_DIRECTION_ID
FIELD_INTERFACE_NAME
FIELD_NW_PROTOCOL_ID
FIELD_INFO1
FIELD_INFO2
FIELD_INFO3
FIELD_INFO4
FIELD_START_TIME
FIELD_ELAPSED_TIME
FIELD_PACKETS
FIELD_BYTES
FIELD_CLIENT_INBOUND_PACKETS
FIELD_CLIENT_OUTBOUND_PACKETS
FIELD_SERVER_INBOUND_PACKETS
FIELD_SERVER_OUTBOUND_PACKETS
FIELD_CLIENT_INBOUND_BYTES
FIELD_CLIENT_OUTBOUND_BYTES
FIELD_SERVER_INBOUND_BYTES
FIELD_SERVER_OUTBOUND_BYTES
26500
27500
Memory
27100

Fig. 27B_3


FIELD_SOURCE_HOST_NAME
FIELD_DESTINATION_HOST_NAME
FIELD_SOURCE_SERVICE_NAME
FIELD_DESTINATION_SERVICE_NAME
Memory
27100

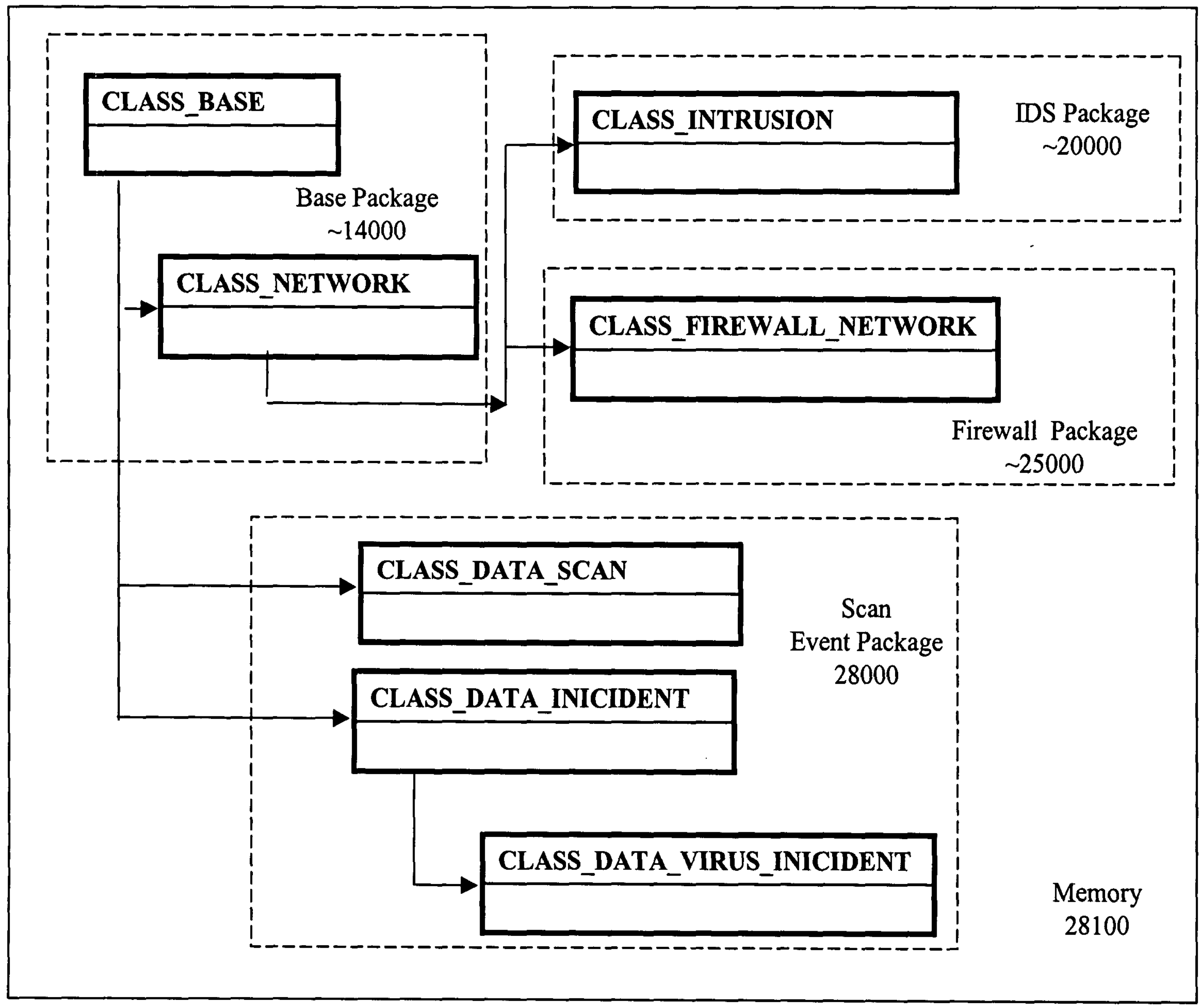

CLASS_BASE
Base Package
~14000
CLASS_NETWORK
CLASS_INTRUSION
IDS Package
~20000
CLASS_FIREWALL_NETWORK
Firewall Package
~25000
CLASS_DATA_SCAN
Scan
Event Package
28000
CLASS_DATA_INICIDENT
CLASS_DATA_VIRUS_INICIDENT
Memory
28100

FIG. 28

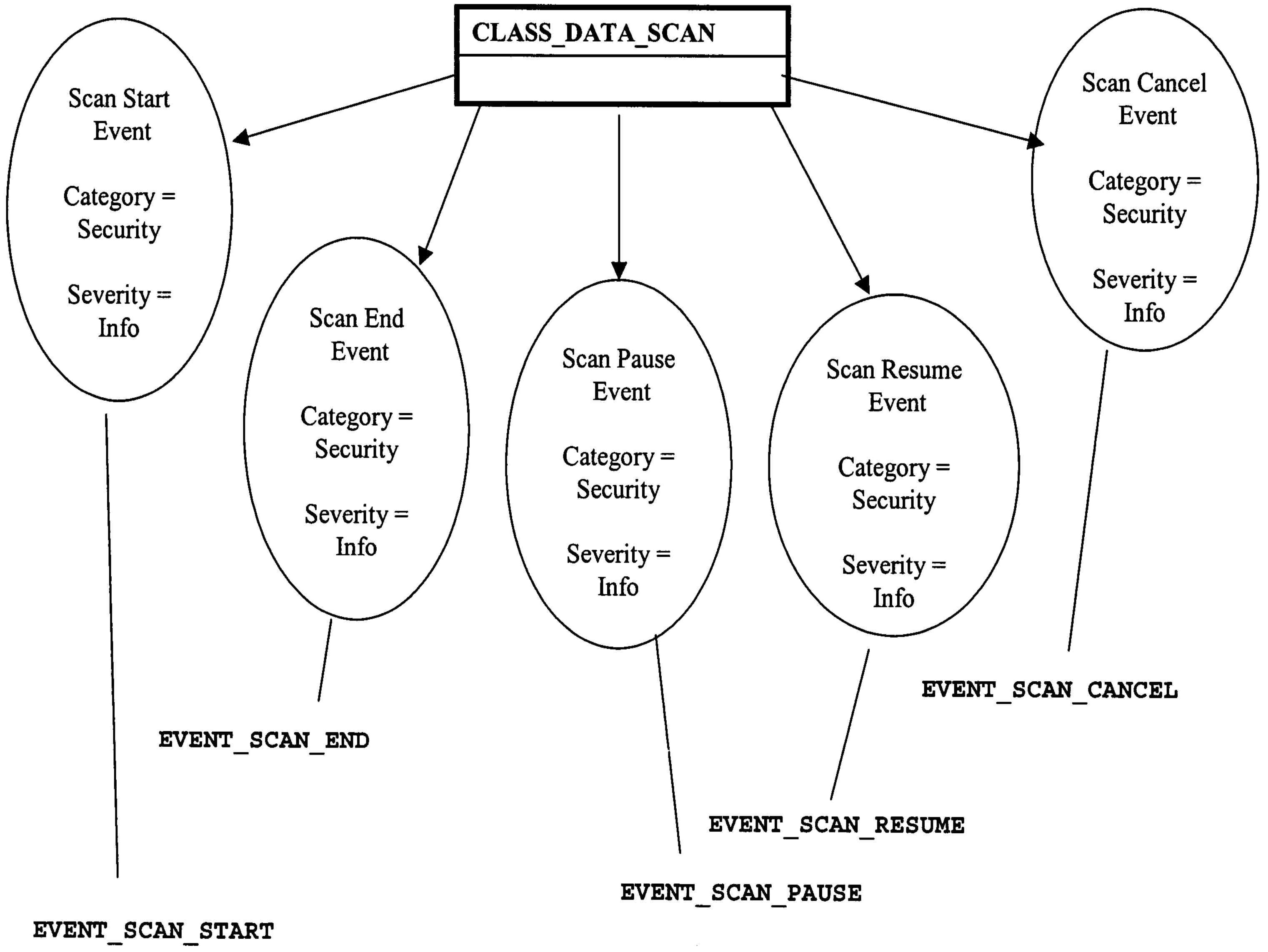

CLASS_DATA_SCAN
Scan Start Event
Category = Security
Severity = Info
Scan End Event
Category = Security
Severity = Info
Scan Pause Event
Category = Security
Severity = Info
Scan Resume Event
Category = Security
Severity = Info
Scan Cancel Event
Category = Security
Severity = Info
EVENT_SCAN_START
EVENT_SCAN_END
EVENT_SCAN_PAUSE
EVENT_SCAN_RESUME
EVENT_SCAN_CANCEL

FIG. 29A

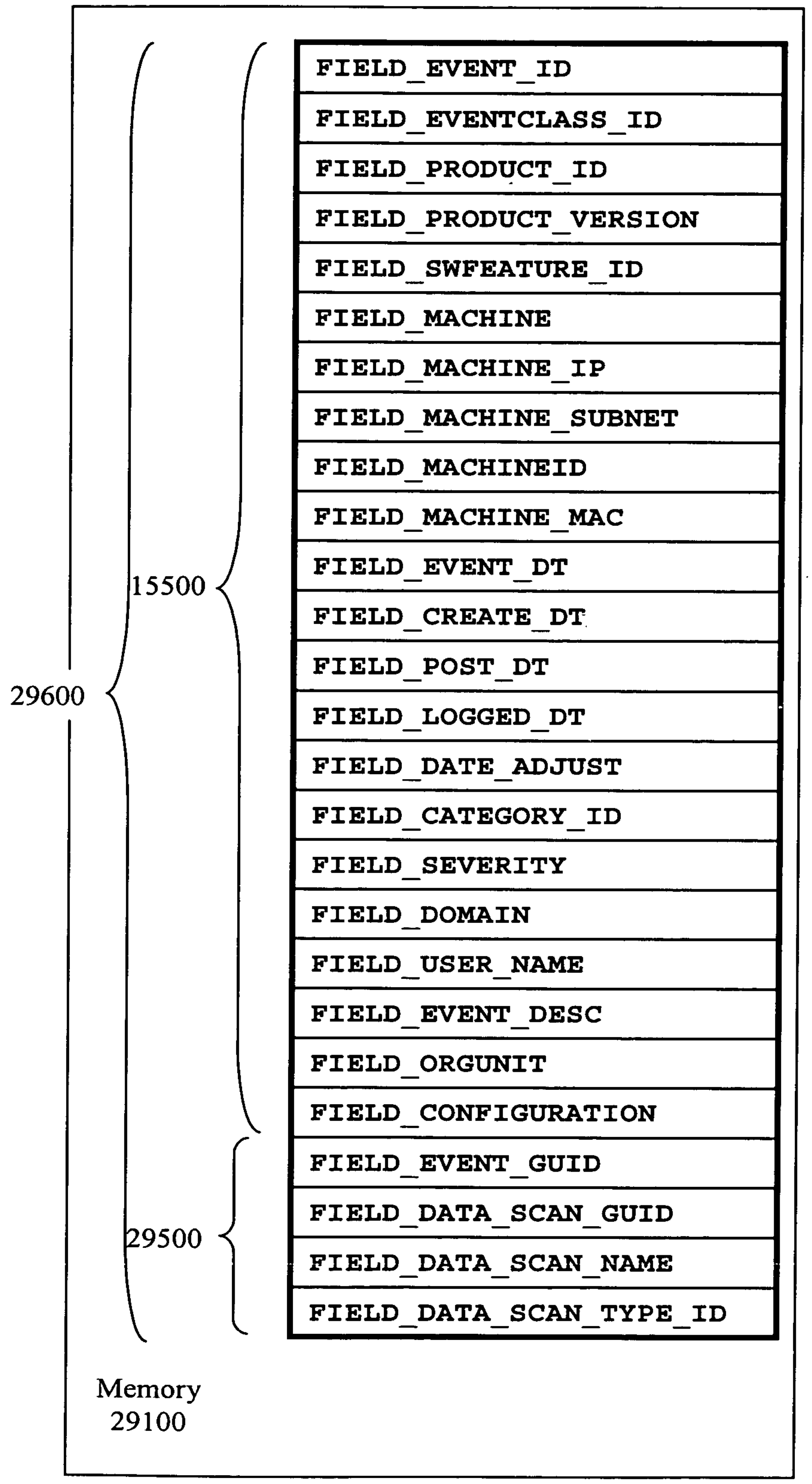

FIELD_EVENT_ID
FIELD_EVENTCLASS_ID
FIELD_PRODUCT_ID
FIELD_PRODUCT_VERSION
FIELD_SWFEATURE_ID
FIELD_MACHINE
FIELD_MACHINE_IP
FIELD_MACHINE_SUBNET
FIELD_MACHINEID
FIELD_MACHINE_MAC
FIELD_EVENT_DT
FIELD_CREATE_DT
FIELD_POST_DT
FIELD_LOGGED_DT
FIELD_DATE_ADJUST
FIELD_CATEGORY_ID
FIELD_SEVERITY
FIELD_DOMAIN
FIELD_USER_NAME
FIELD_EVENT_DESC
FIELD_ORGUNIT
FIELD_CONFIGURATION
FIELD_EVENT_GUID
FIELD_DATA_SCAN_GUID
FIELD_DATA_SCAN_NAME
FIELD_DATA_SCAN_TYPE_ID
15500
29600
29500
Memory
29100

Fig. 29B

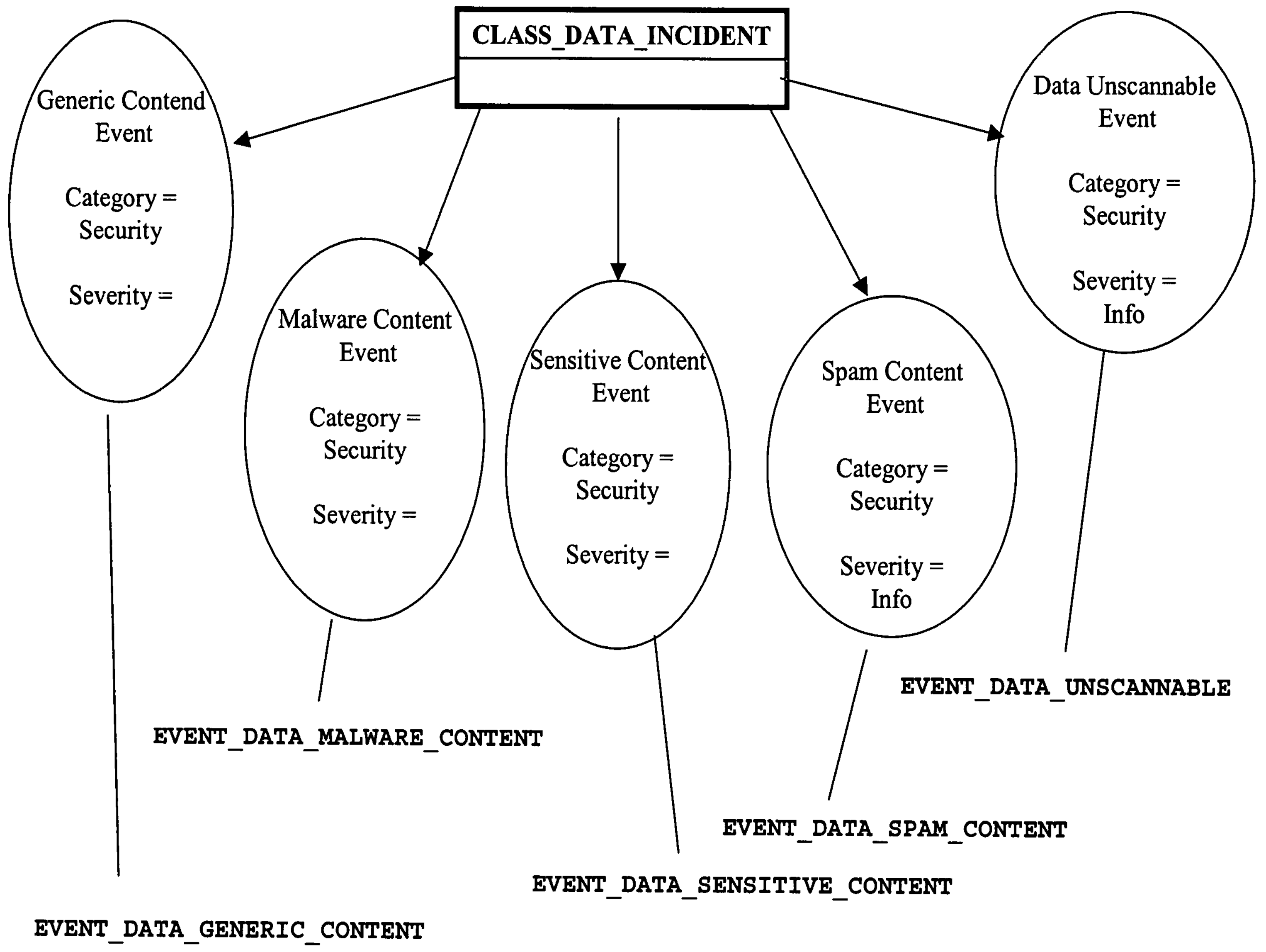

CLASS_DATA_INCIDENT
Generic Contend Event
Category = Security
Severity =
Malware Content Event
Category = Security
Severity =
Sensitive Content Event
Category = Security
Severity =
Spam Content Event
Category = Security
Severity = Info
Data Unscannable Event
Category = Security
Severity = Info
EVENT_DATA_UNSCANNABLE
EVENT_DATA_MALWARE_CONTENT
EVENT_DATA_SPAM_CONTENT
EVENT_DATA_SENSITIVE_CONTENT
EVENT_DATA_GENERIC_CONTENT

FIG. 30A

Fig. 30B_1

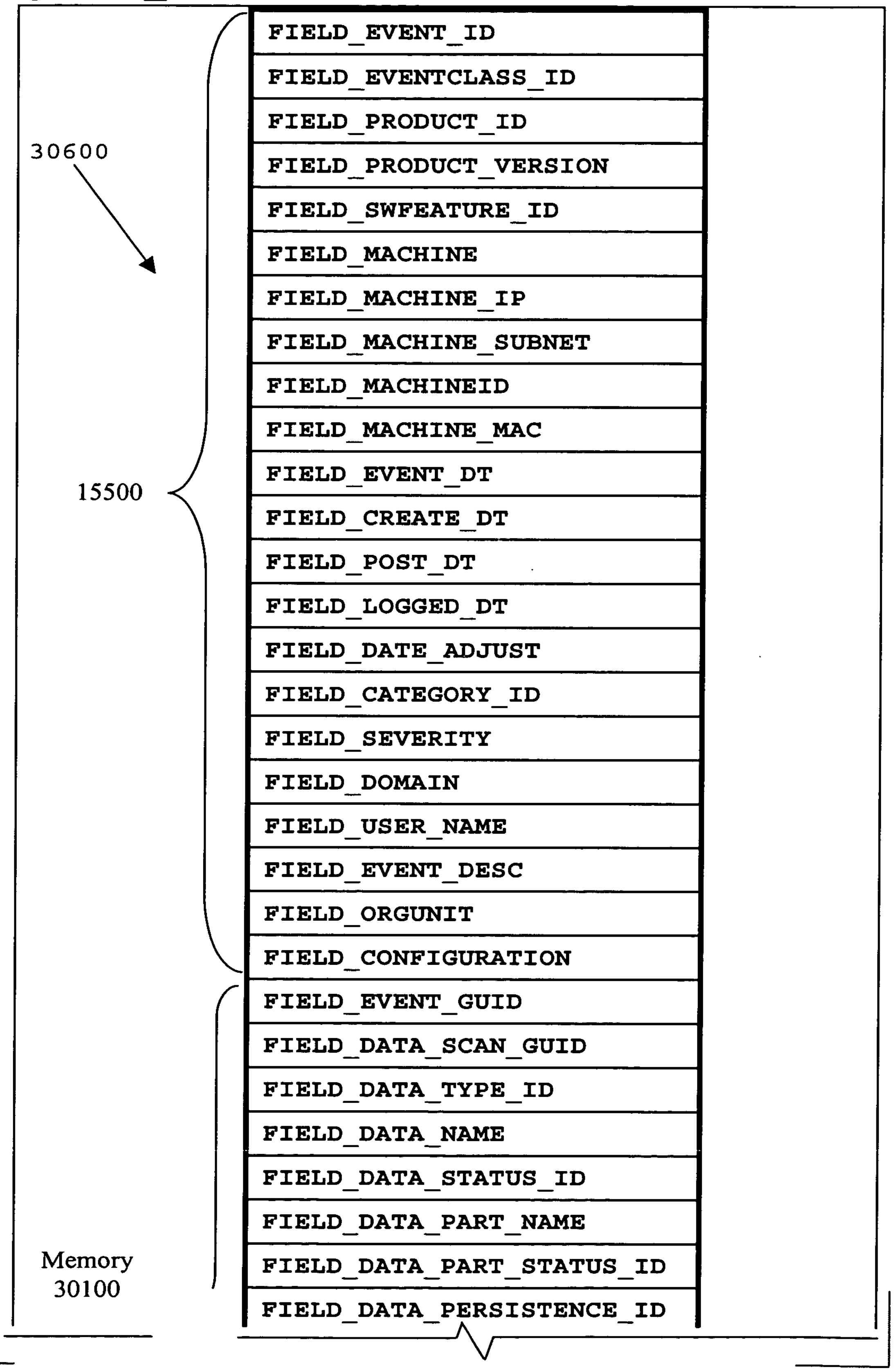
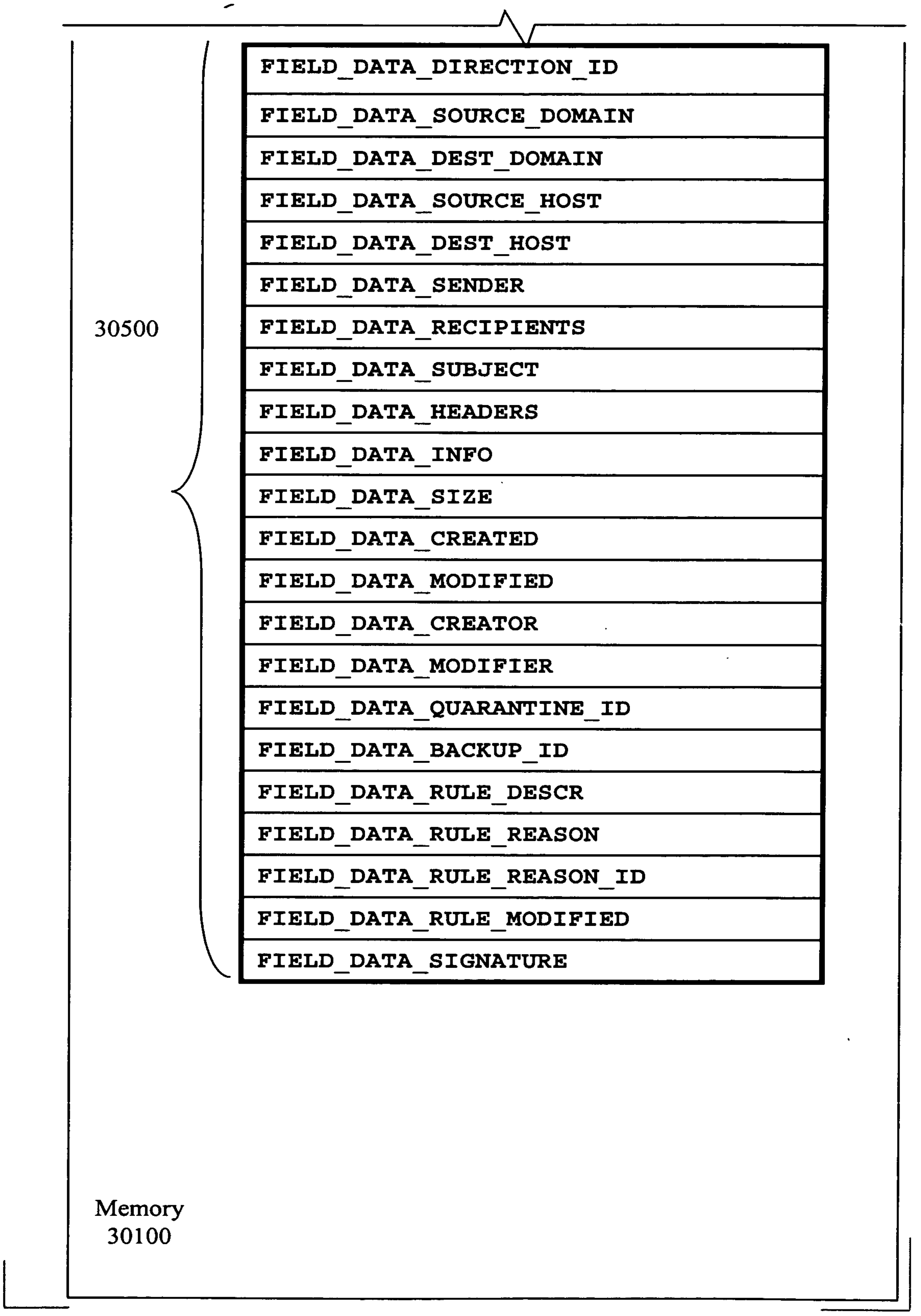

30600
15500
Memory
30100
FIELD_EVENT_ID
FIELD_EVENTCLASS_ID
FIELD_PRODUCT_ID
FIELD_PRODUCT_VERSION
FIELD_SWFEATURE_ID
FIELD_MACHINE
FIELD_MACHINE_IP
FIELD_MACHINE_SUBNET
FIELD_MACHINEID
FIELD_MACHINE_MAC
FIELD_EVENT_DT
FIELD_CREATE_DT
FIELD_POST_DT
FIELD_LOGGED_DT
FIELD_DATE_ADJUST
FIELD_CATEGORY_ID
FIELD_SEVERITY
FIELD_DOMAIN
FIELD_USER_NAME
FIELD_EVENT_DESC
FIELD_ORGUNIT
FIELD_CONFIGURATION
FIELD_EVENT_GUID
FIELD_DATA_SCAN_GUID
FIELD_DATA_TYPE_ID
FIELD_DATA_NAME
FIELD_DATA_STATUS_ID
FIELD_DATA_PART_NAME
FIELD_DATA_PART_STATUS_ID
FIELD_DATA_PERSISTENCE_ID

Fig. 30B_2


FIELD_DATA_DIRECTION_ID
FIELD_DATA_SOURCE_DOMAIN
FIELD_DATA_DEST_DOMAIN
FIELD_DATA_SOURCE_HOST
FIELD_DATA_DEST_HOST
FIELD_DATA_SENDER
FIELD_DATA_RECIPIENTS
FIELD_DATA_SUBJECT
FIELD_DATA_HEADERS
FIELD_DATA_INFO
FIELD_DATA_SIZE
FIELD_DATA_CREATED
FIELD_DATA_MODIFIED
FIELD_DATA_CREATOR
FIELD_DATA_MODIFIER
FIELD_DATA_QUARANTINE_ID
FIELD_DATA_BACKUP_ID
FIELD_DATA_RULE_DESCR
FIELD_DATA_RULE_REASON
FIELD_DATA_RULE_REASON_ID
FIELD_DATA_RULE_MODIFIED
FIELD_DATA_SIGNATURE
30500
Memory
30100

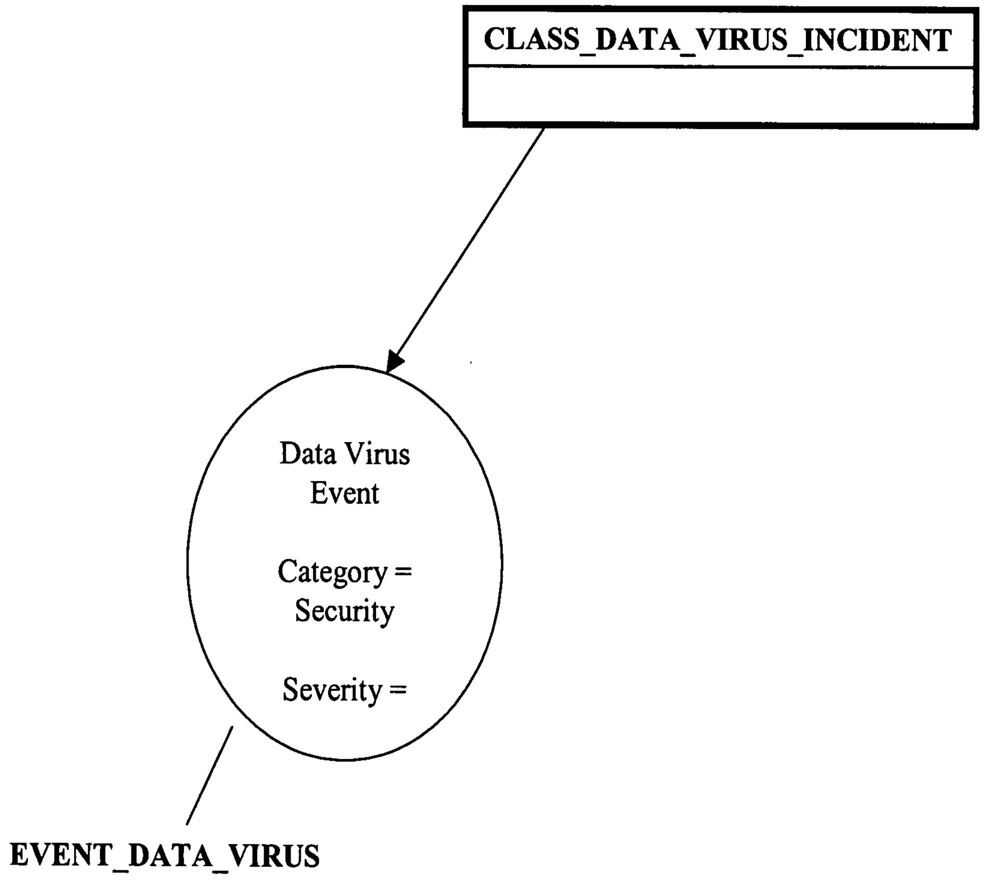
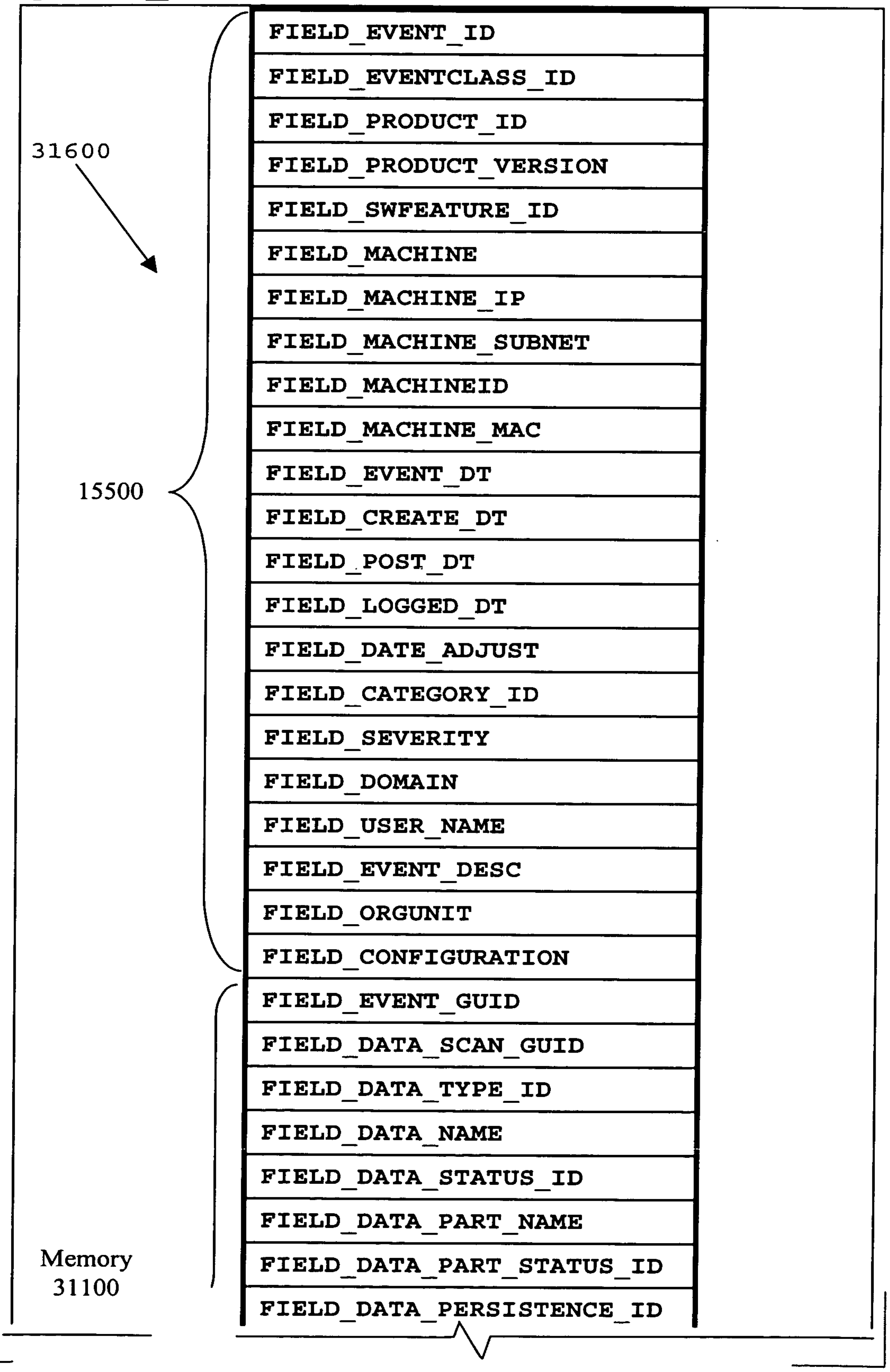

CLASS_DATA_VIRUS_INCIDENT
Data Virus Event
Category = Security
Severity =
EVENT_DATA_VIRUS

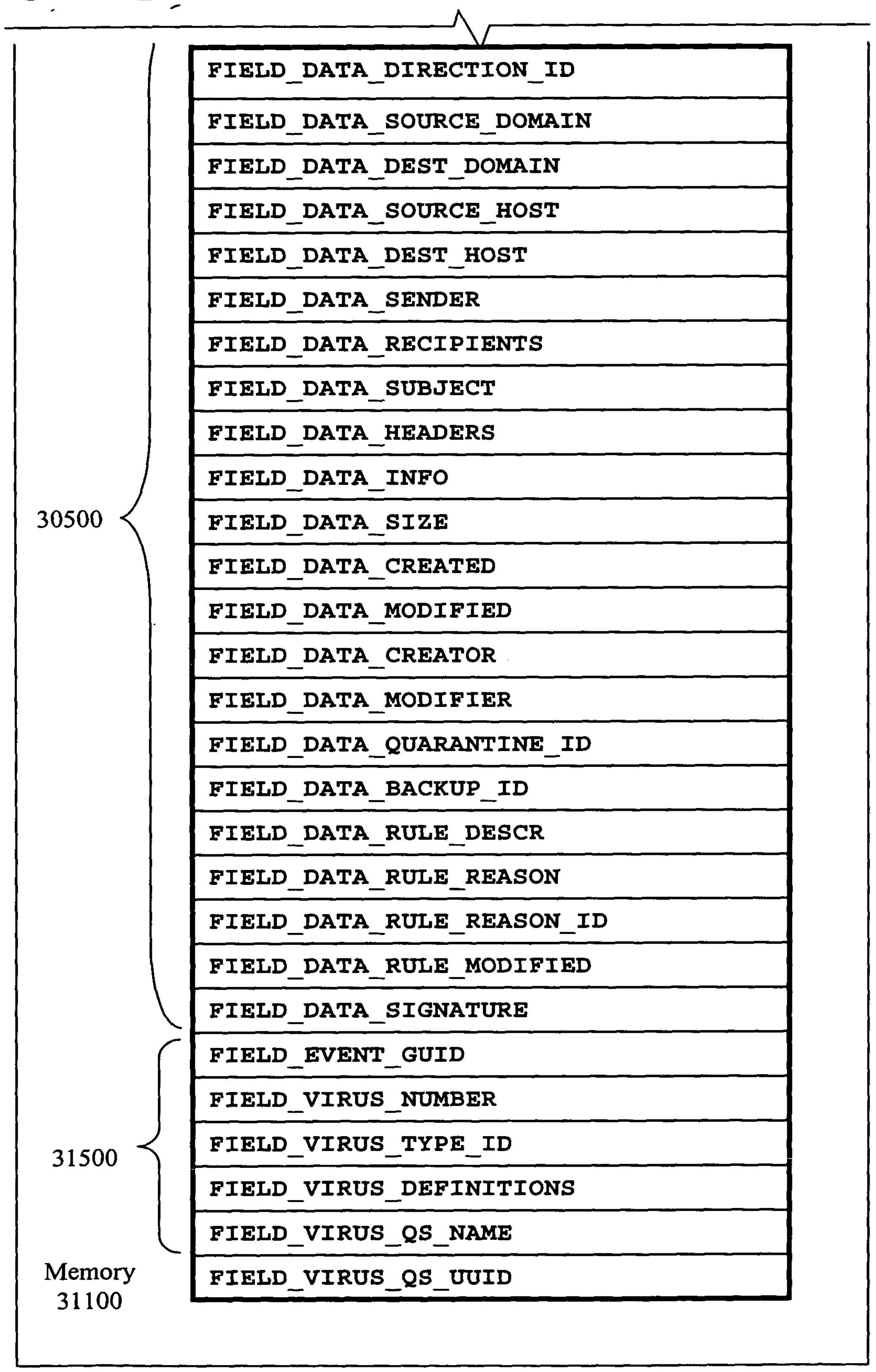
FIG. 31A

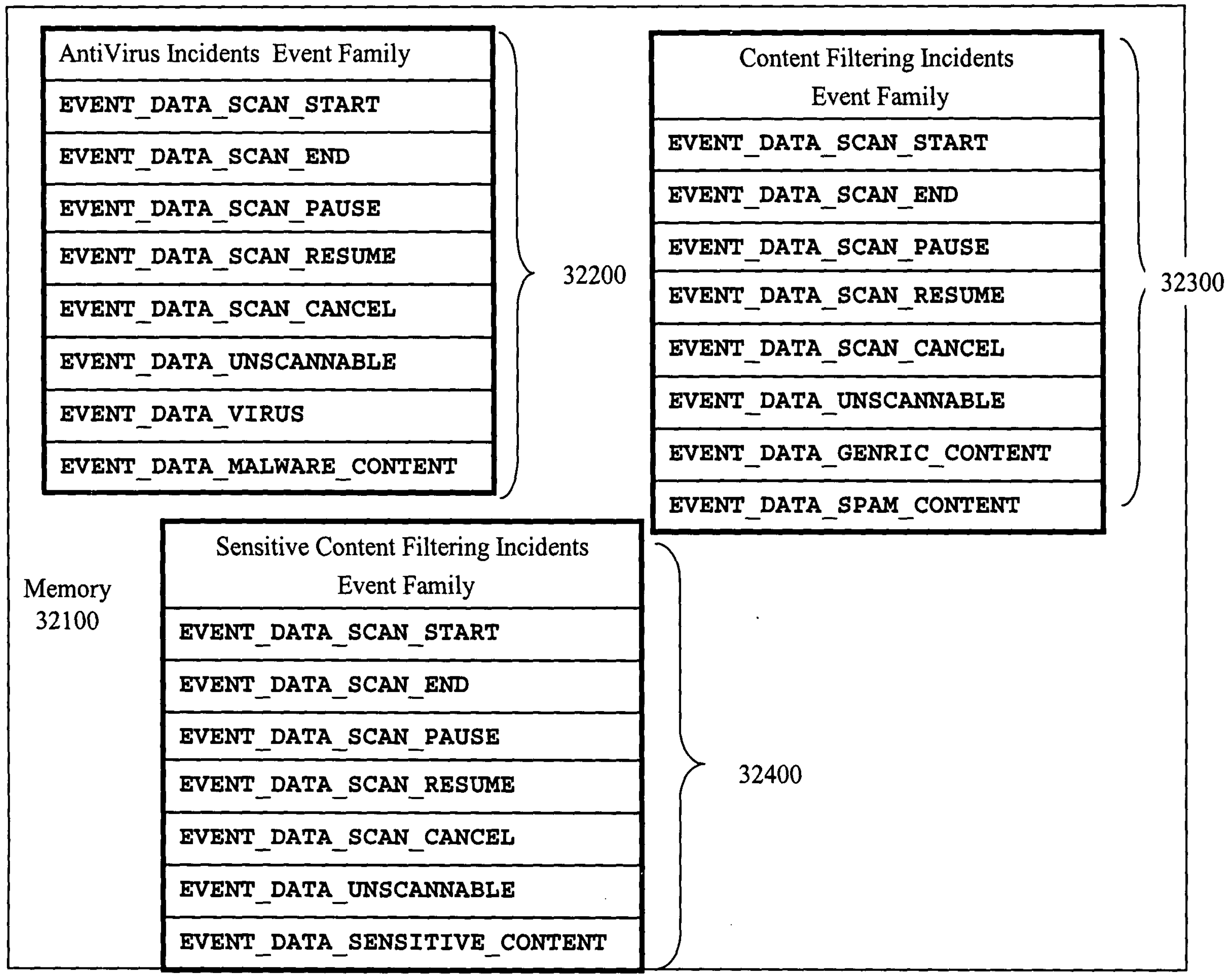
Fig. 31B_1



31600
15500
FIELD_EVENT_ID
FIELD_EVENTCLASS_ID
FIELD_PRODUCT_ID
FIELD_PRODUCT_VERSION
FIELD_SWFEATURE_ID
FIELD_MACHINE
FIELD_MACHINE_IP
FIELD_MACHINE_SUBNET
FIELD_MACHINEID
FIELD_MACHINE_MAC
FIELD_EVENT_DT
FIELD_CREATE_DT
FIELD_POST_DT
FIELD_LOGGED_DT
FIELD_DATE_ADJUST
FIELD_CATEGORY_ID
FIELD_SEVERITY
FIELD_DOMAIN
FIELD_USER_NAME
FIELD_EVENT_DESC
FIELD_ORGUNIT
FIELD_CONFIGURATION
FIELD_EVENT_GUID
FIELD_DATA_SCAN_GUID
FIELD_DATA_TYPE_ID
FIELD_DATA_NAME
FIELD_DATA_STATUS_ID
FIELD_DATA_PART_NAME
FIELD_DATA_PART_STATUS_ID
FIELD_DATA_PERSISTENCE_ID
Memory
31100

Fig. 31B_2

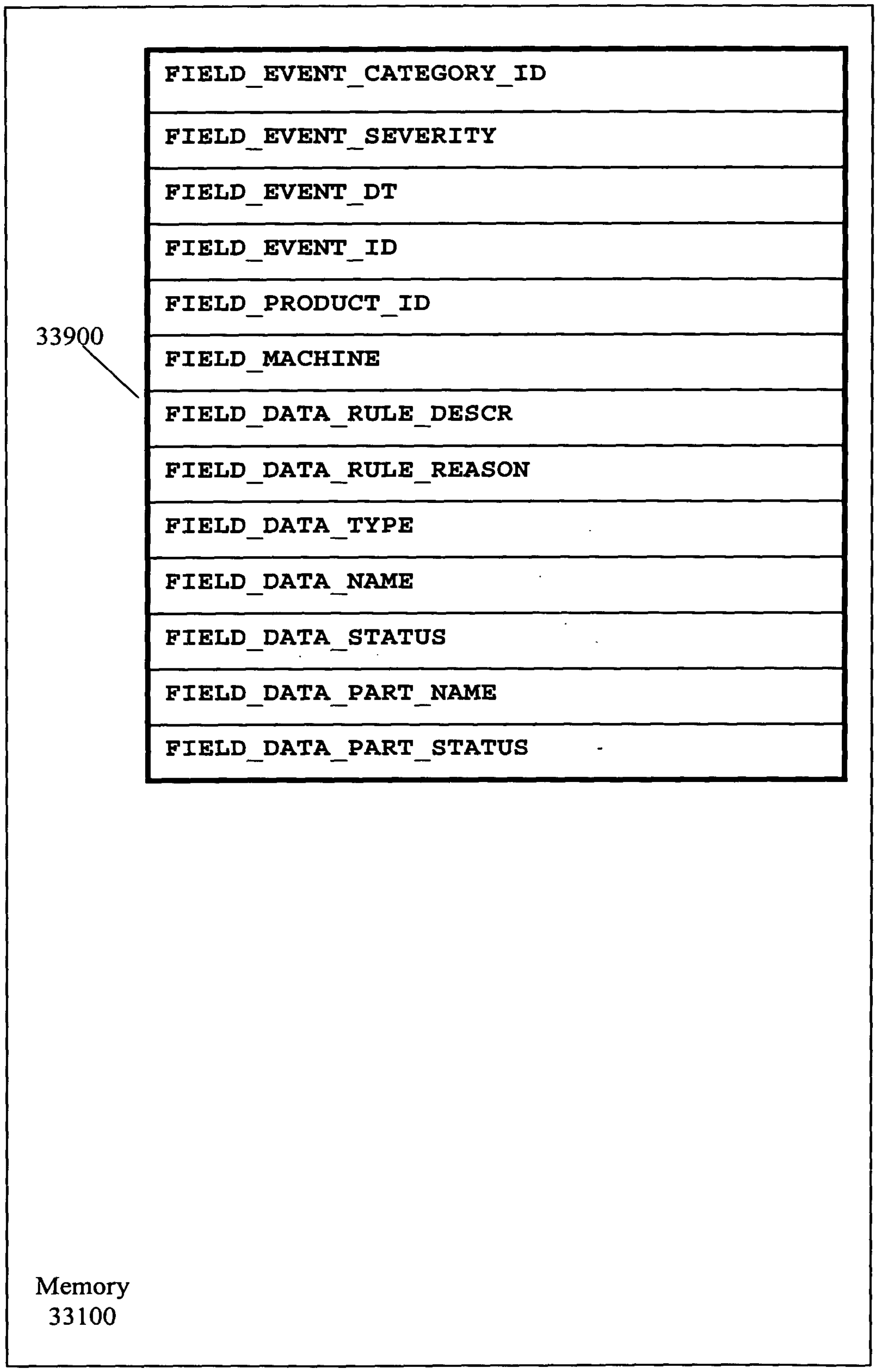

FIELD_DATA_DIRECTION_ID
FIELD_DATA_SOURCE_DOMAIN
FIELD_DATA_DEST_DOMAIN
FIELD_DATA_SOURCE_HOST
FIELD_DATA_DEST_HOST
FIELD_DATA_SENDER
FIELD_DATA_RECIPIENTS
FIELD_DATA_SUBJECT
FIELD_DATA_HEADERS
FIELD_DATA_INFO
FIELD_DATA_SIZE
FIELD_DATA_CREATED
FIELD_DATA_MODIFIED
FIELD_DATA_CREATOR
FIELD_DATA_MODIFIER
FIELD_DATA_QUARANTINE_ID
FIELD_DATA_BACKUP_ID
FIELD_DATA_RULE_DESCR
FIELD_DATA_RULE_REASON
FIELD_DATA_RULE_REASON_ID
FIELD_DATA_RULE_MODIFIED
FIELD_DATA_SIGNATURE
FIELD_EVENT_GUID
FIELD_VIRUS_NUMBER
FIELD_VIRUS_TYPE_ID
FIELD_VIRUS_DEFINITIONS
FIELD_VIRUS_QS_NAME
FIELD_VIRUS_QS_UUID
30500
31500
Memory
31100



AntiVirus Incidents Event Family
EVENT_DATA_SCAN_START
EVENT_DATA_SCAN_END
EVENT_DATA_SCAN_PAUSE
EVENT_DATA_SCAN_RESUME
EVENT_DATA_SCAN_CANCEL
EVENT_DATA_UNSCANNABLE
EVENT_DATA_VIRUS
EVENT_DATA_MALWARE_CONTENT
32200
Content Filtering Incidents Event Family
EVENT_DATA_SCAN_START
EVENT_DATA_SCAN_END
EVENT_DATA_SCAN_PAUSE
EVENT_DATA_SCAN_RESUME
EVENT_DATA_SCAN_CANCEL
EVENT_DATA_UNSCANNABLE
EVENT_DATA_GENRIC_CONTENT
EVENT_DATA_SPAM_CONTENT
32300
Memory 32100
Sensitive Content Filtering Incidents Event Family
EVENT_DATA_SCAN_START
EVENT_DATA_SCAN_END
EVENT_DATA_SCAN_PAUSE
EVENT_DATA_SCAN_RESUME
EVENT_DATA_SCAN_CANCEL
EVENT_DATA_UNSCANNABLE
EVENT_DATA_SENSITIVE_CONTENT
32400

FIG. 32

Fig. 33



FIELD_EVENT_CATEGORY_ID
FIELD_EVENT_SEVERITY
FIELD_EVENT_DT
FIELD_EVENT_ID
FIELD_PRODUCT_ID
FIELD_MACHINE
FIELD_DATA_RULE_DESCR
FIELD_DATA_RULE_REASON
FIELD_DATA_TYPE
FIELD_DATA_NAME
FIELD_DATA_STATUS
FIELD_DATA_PART_NAME
FIELD_DATA_PART_STATUS

Fig. 34

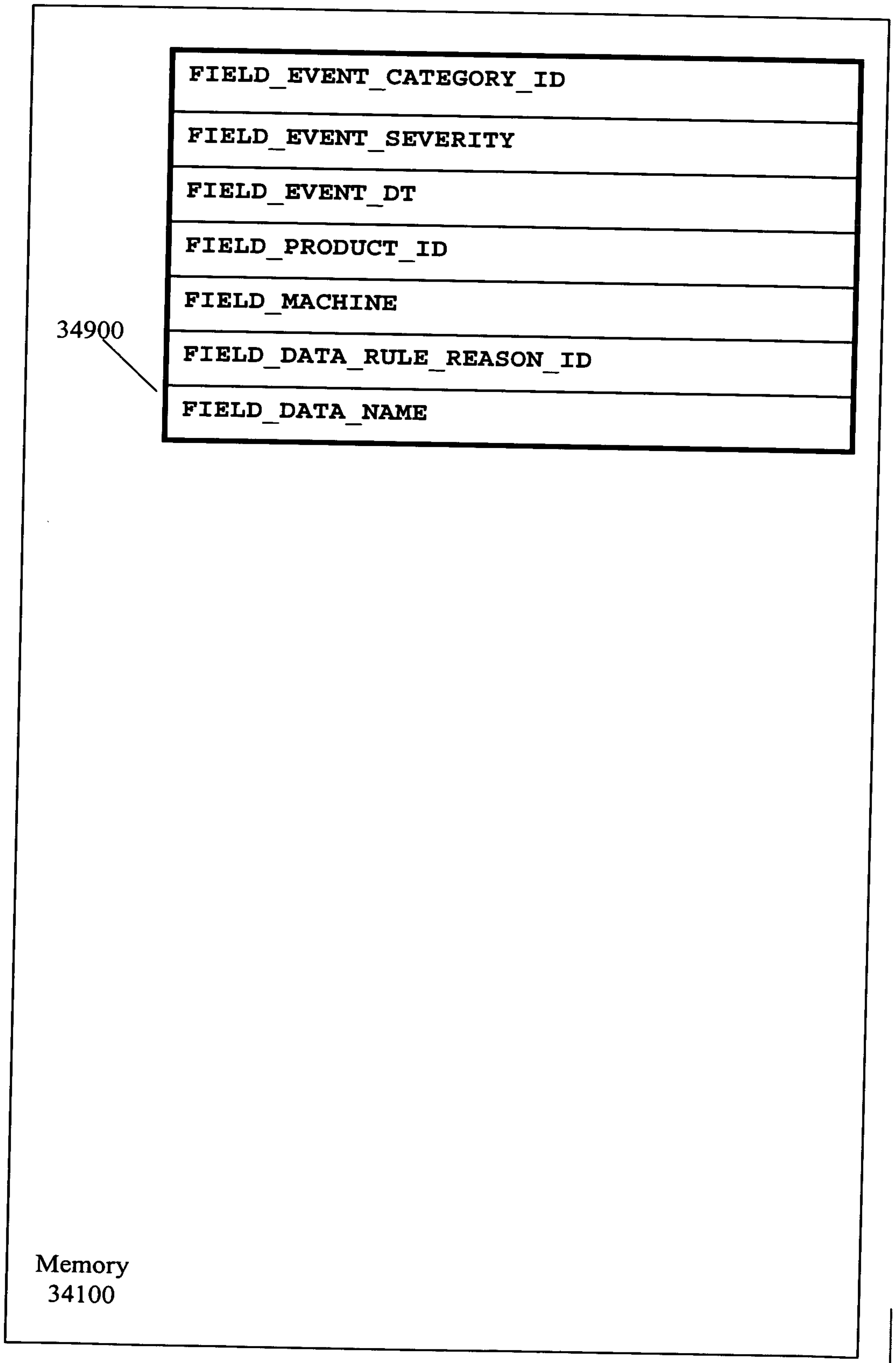
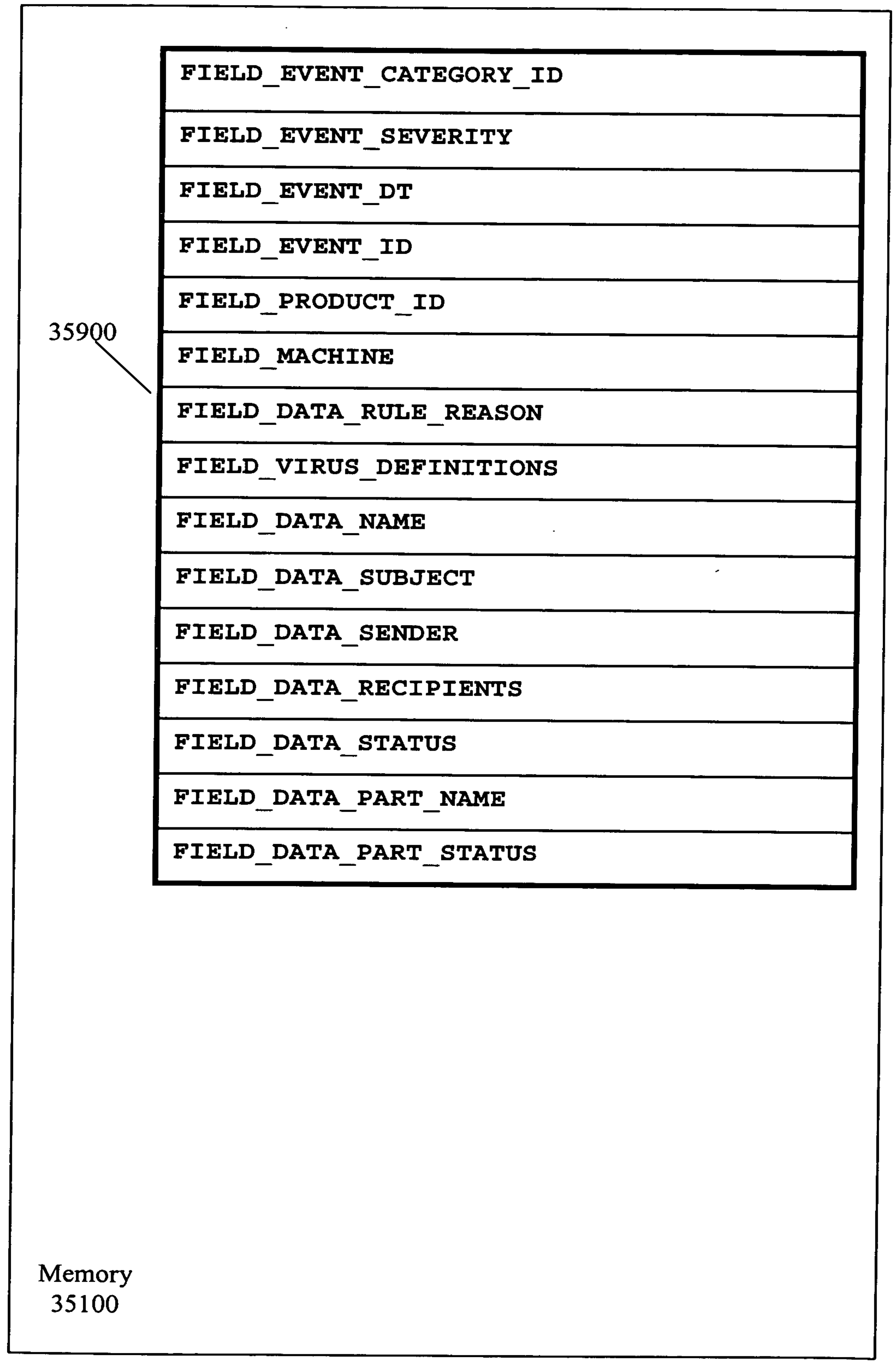
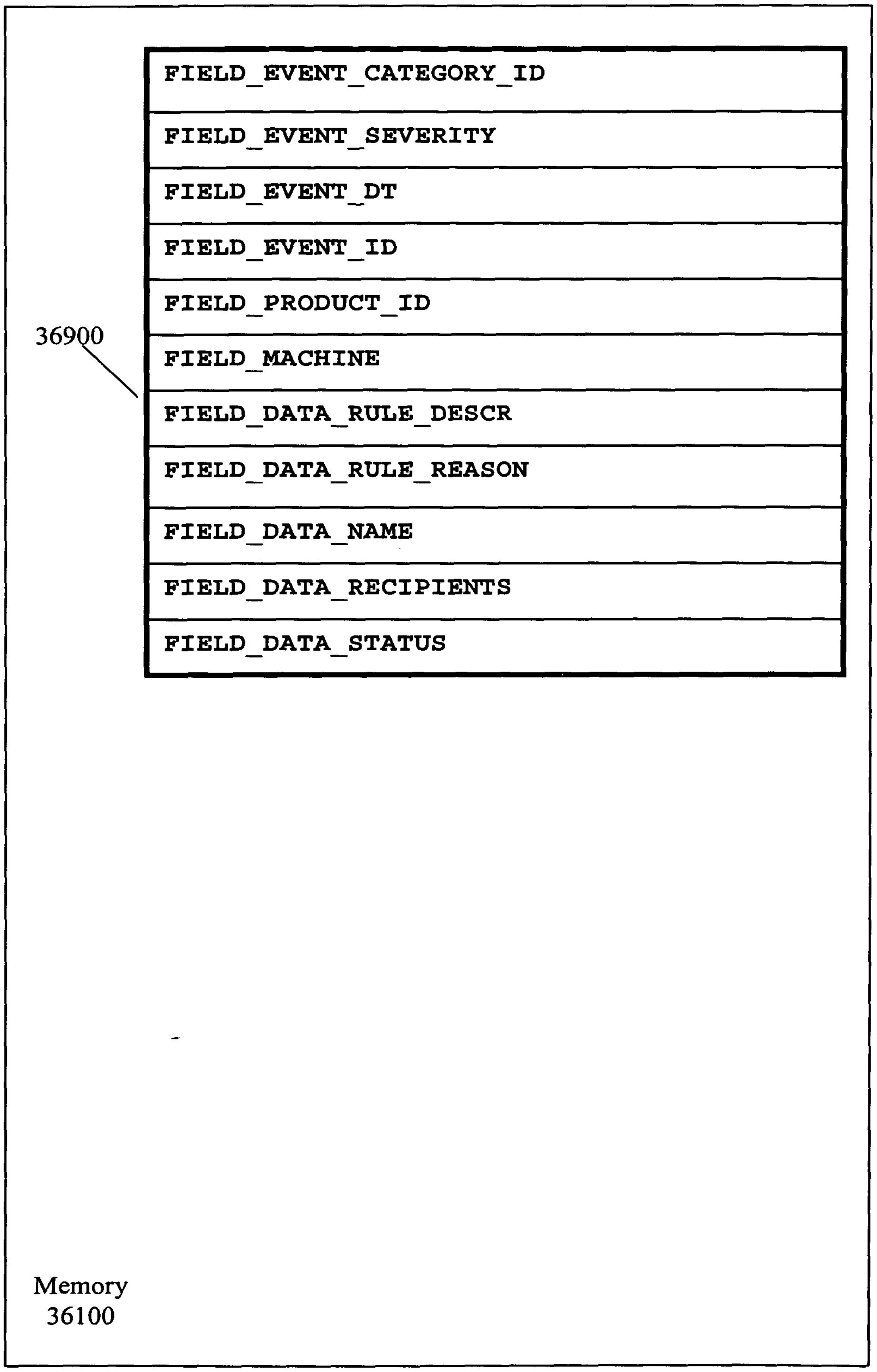

FIELD_EVENT_CATEGORY_ID
FIELD_EVENT_SEVERITY
FIELD_EVENT_DT
FIELD_PRODUCT_ID
FIELD_MACHINE
FIELD_DATA_RULE_REASON_ID
FIELD_DATA_NAME
34900
Memory
34100

Fig. 35



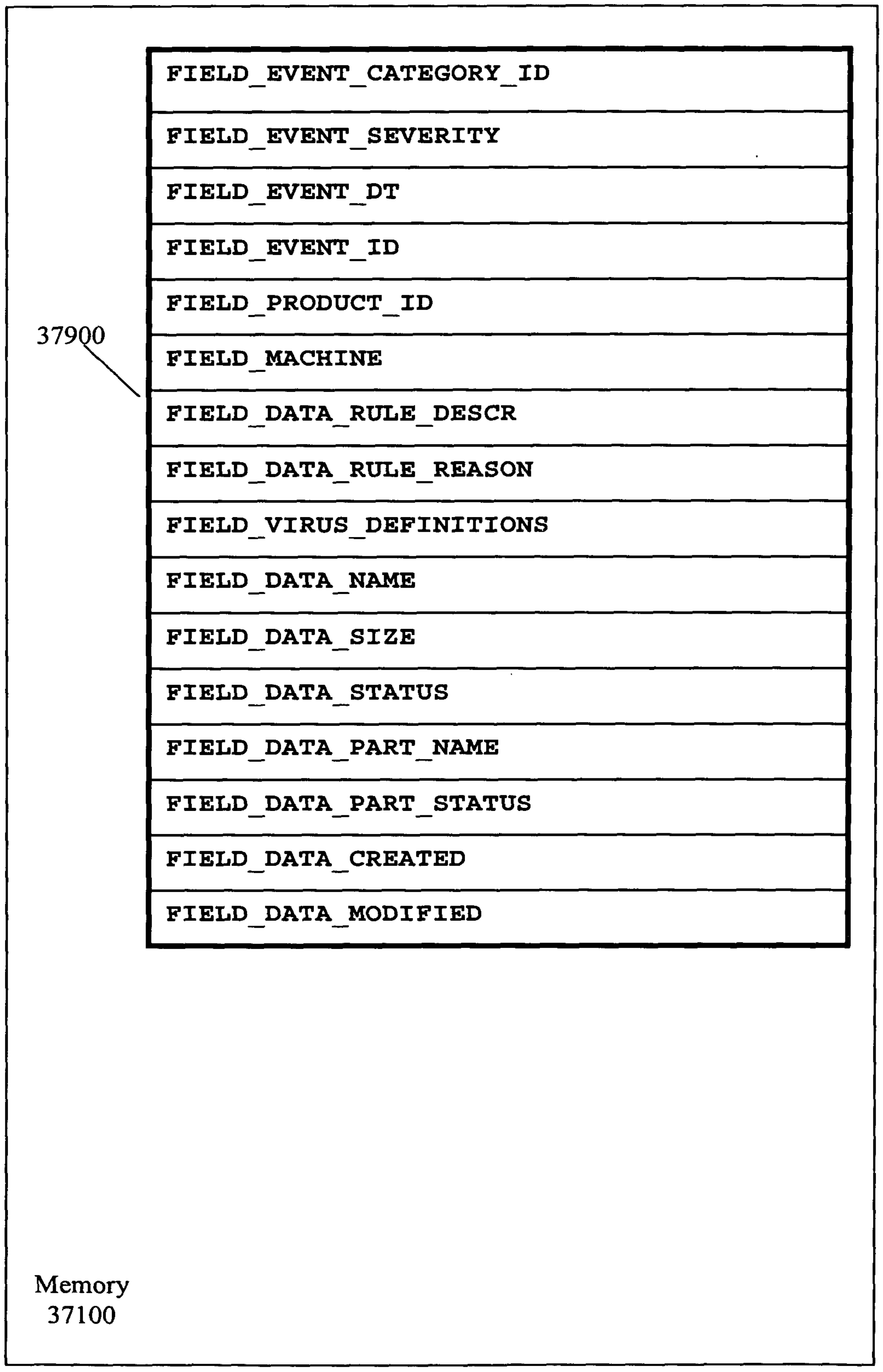
FIELD_EVENT_CATEGORY_ID
FIELD_EVENT_SEVERITY
FIELD_EVENT_DT
FIELD_EVENT_ID
FIELD_PRODUCT_ID
FIELD_MACHINE
FIELD_DATA_RULE_REASON
FIELD_VIRUS_DEFINITIONS
FIELD_DATA_NAME
FIELD_DATA_SUBJECT
FIELD_DATA_SENDER
FIELD_DATA_RECIPIENTS
FIELD_DATA_STATUS
FIELD_DATA_PART_NAME
FIELD_DATA_PART_STATUS
35900
Memory
35100

Fig. 36



FIELD_EVENT_CATEGORY_ID
FIELD_EVENT_SEVERITY
FIELD_EVENT_DT
FIELD_EVENT_ID
FIELD_PRODUCT_ID
FIELD_MACHINE
FIELD_DATA_RULE_DESCR
FIELD_DATA_RULE_REASON
FIELD_DATA_NAME
FIELD_DATA_RECIPIENTS
FIELD_DATA_STATUS
36900
Memory
36100

Fig. 37


FIELD_EVENT_CATEGORY_ID
FIELD_EVENT_SEVERITY
FIELD_EVENT_DT
FIELD_EVENT_ID
FIELD_PRODUCT_ID
FIELD_MACHINE
FIELD_DATA_RULE_DESCR
FIELD_DATA_RULE_REASON
FIELD_VIRUS_DEFINITIONS
FIELD_DATA_NAME
FIELD_DATA_SIZE
FIELD_DATA_STATUS
FIELD_DATA_PART_NAME
FIELD_DATA_PART_STATUS
FIELD_DATA_CREATED
FIELD_DATA_MODIFIED
37900
Memory
37100

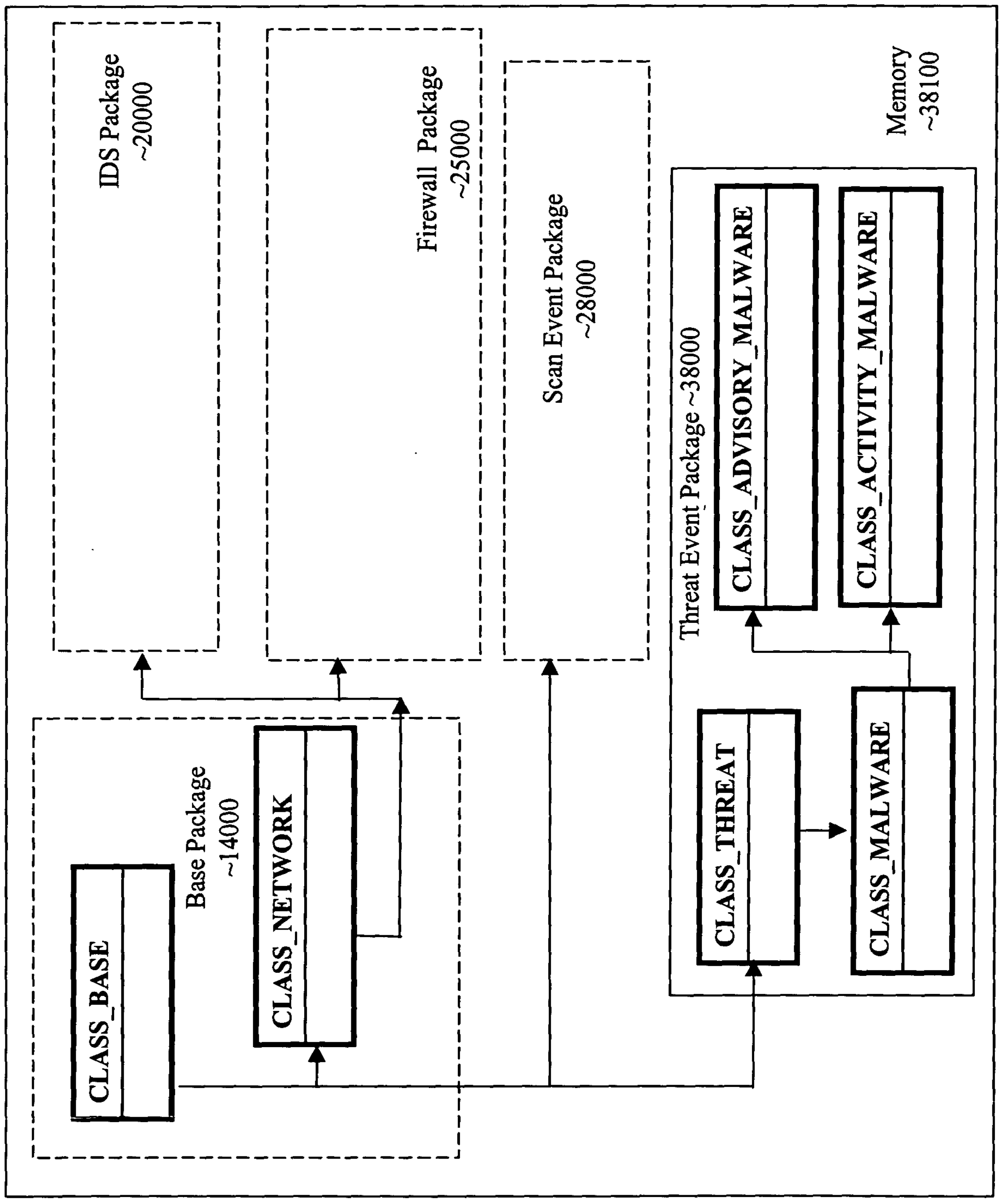

IDS Package
~20000
Firewall Package
~25000
Scan Event Package
~28000
Threat Event Package ~38000
Memory
~38100
CLASS_ADVISORY_MALWARE
CLASS_ACTIVITY_MALWARE
Base Package
~14000
CLASS_BASE
CLASS_NETWORK
CLASS_THREAT
CLASS_MALWARE

FIG. 38

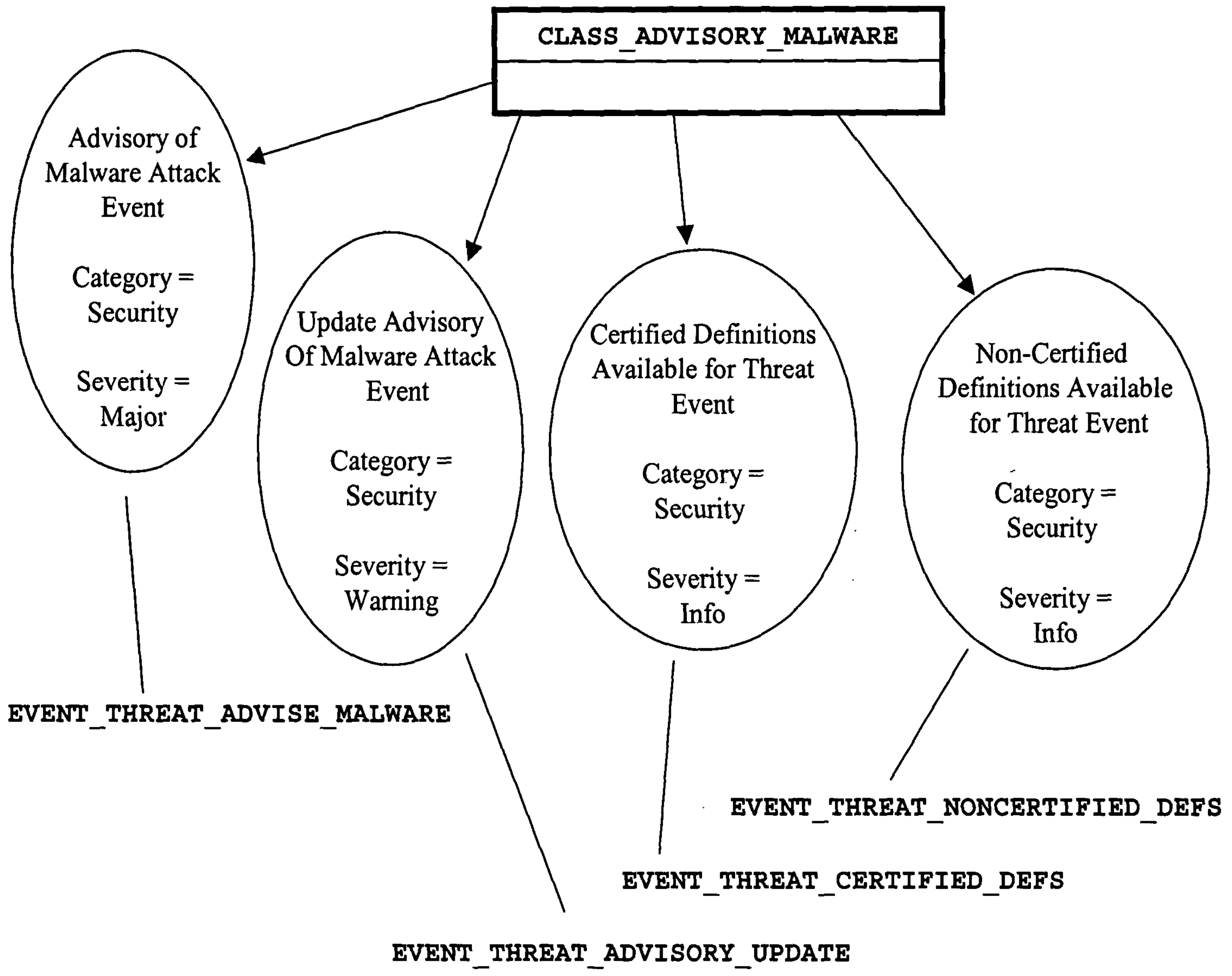

CLASS_ADVISORY_MALWARE
Advisory of Malware Attack Event
Category = Security
Severity = Major
Update Advisory Of Malware Attack Event
Category = Security
Severity = Warning
Certified Definitions Available for Threat Event
Category = Security
Severity = Info
Non-Certified Definitions Available for Threat Event
Category = Security
Severity = Info
EVENT_THREAT_ADVISE_MALWARE
EVENT_THREAT_NONCERTIFIED_DEFS
EVENT_THREAT_CERTIFIED_DEFS
EVENT_THREAT_ADVISORY_UPDATE

FIG. 39A

Fig. 39B_1

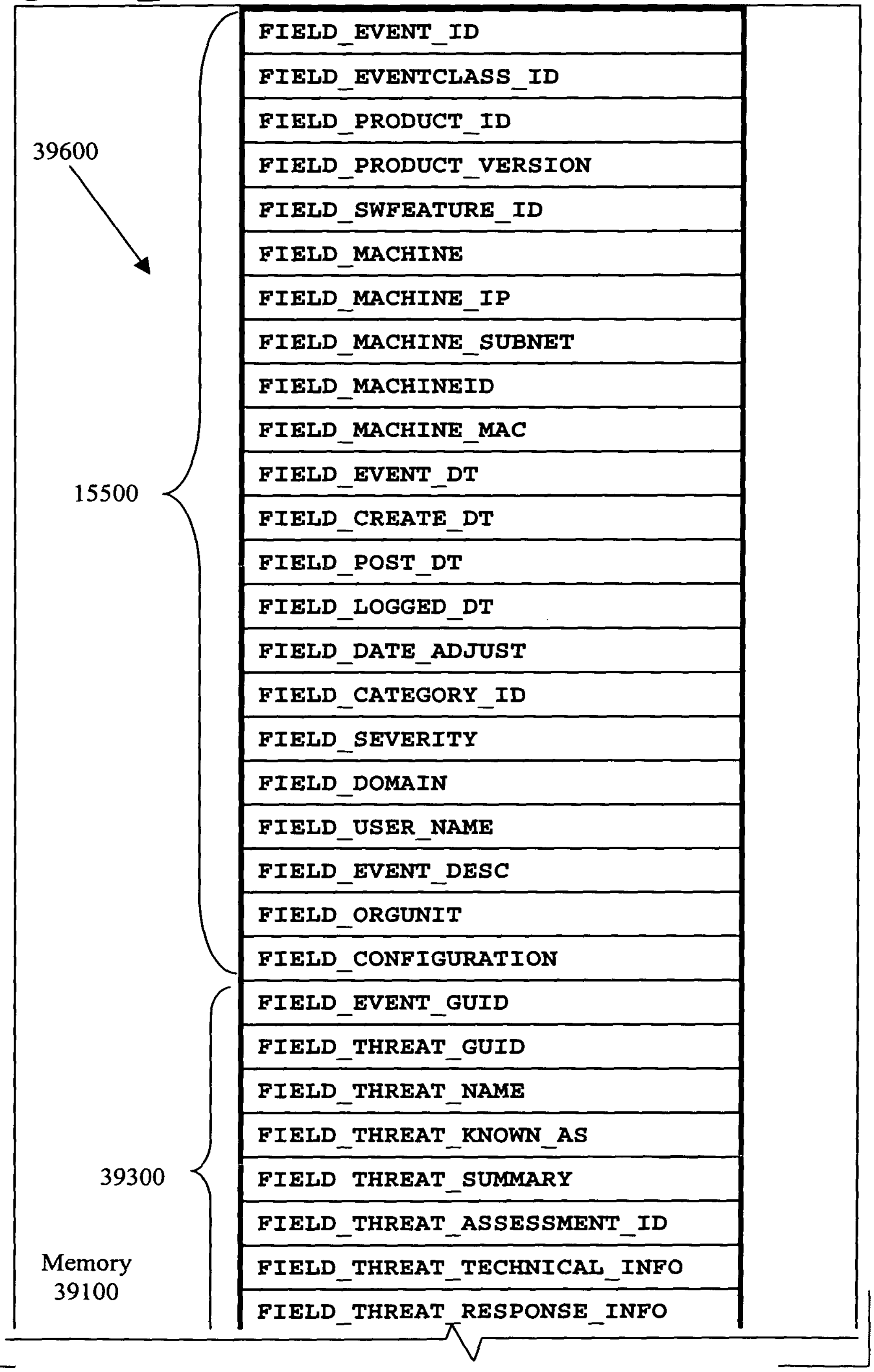
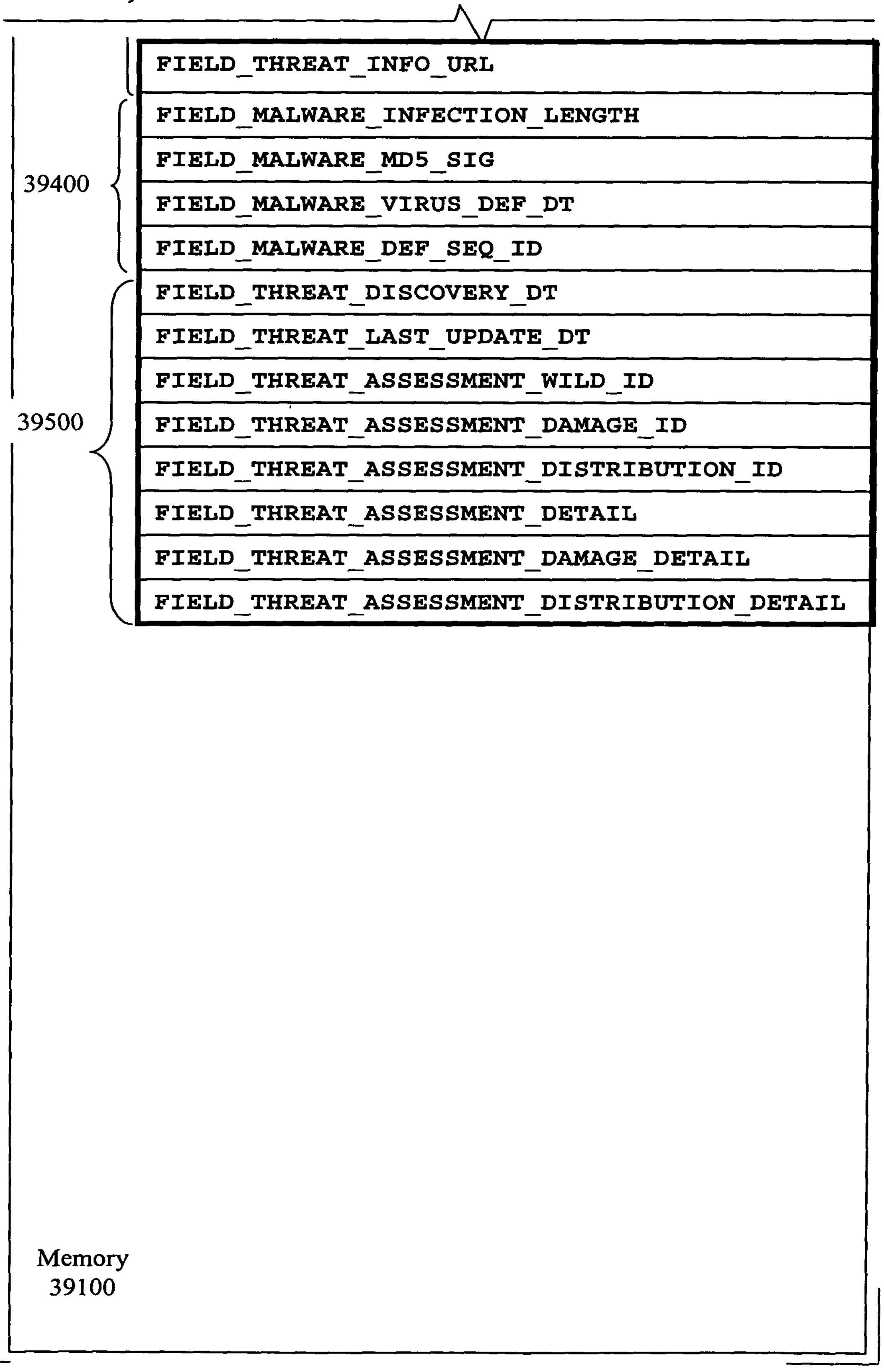

39600
15500
39300
Memory
39100
FIELD_EVENT_ID
FIELD_EVENTCLASS_ID
FIELD_PRODUCT_ID
FIELD_PRODUCT_VERSION
FIELD_SWFEATURE_ID
FIELD_MACHINE
FIELD_MACHINE_IP
FIELD_MACHINE_SUBNET
FIELD_MACHINEID
FIELD_MACHINE_MAC
FIELD_EVENT_DT
FIELD_CREATE_DT
FIELD_POST_DT
FIELD_LOGGED_DT
FIELD_DATE_ADJUST
FIELD_CATEGORY_ID
FIELD_SEVERITY
FIELD_DOMAIN
FIELD_USER_NAME
FIELD_EVENT_DESC
FIELD_ORGUNIT
FIELD_CONFIGURATION
FIELD_EVENT_GUID
FIELD_THREAT_GUID
FIELD_THREAT_NAME
FIELD_THREAT_KNOWN_AS
FIELD THREAT_SUMMARY
FIELD_THREAT_ASSESSMENT_ID
FIELD_THREAT_TECHNICAL_INFO
FIELD_THREAT_RESPONSE_INFO

Fig. 39B_2


FIELD_THREAT_INFO_URL
FIELD_MALWARE_INFECTION_LENGTH
FIELD_MALWARE_MD5_SIG
FIELD_MALWARE_VIRUS_DEF_DT
FIELD_MALWARE_DEF_SEQ_ID
FIELD_THREAT_DISCOVERY_DT
FIELD_THREAT_LAST_UPDATE_DT
FIELD_THREAT_ASSESSMENT_WILD_ID
FIELD_THREAT_ASSESSMENT_DAMAGE_ID
FIELD_THREAT_ASSESSMENT_DISTRIBUTION_ID
FIELD_THREAT_ASSESSMENT_DETAIL
FIELD_THREAT_ASSESSMENT_DAMAGE_DETAIL
FIELD_THREAT_ASSESSMENT_DISTRIBUTION_DETAIL
39400
39500
Memory
39100

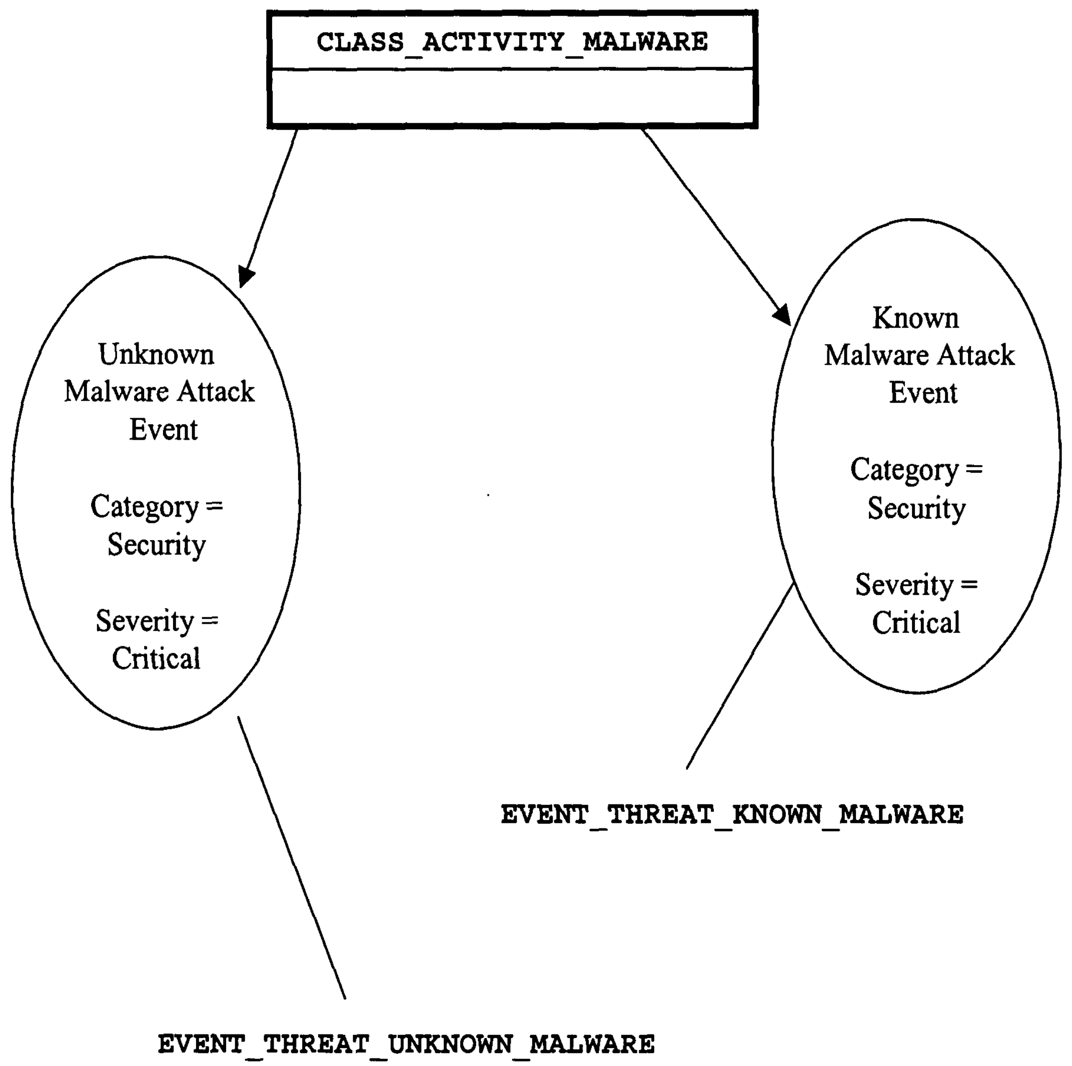

CLASS_ACTIVITY_MALWARE
Unknown Malware Attack Event
Category = Security
Severity = Critical
Known Malware Attack Event
Category = Security
Severity = Critical
EVENT_THREAT_KNOWN_MALWARE
EVENT_THREAT_UNKNOWN_MALWARE

FIG. 40A

Fig. 40B_1

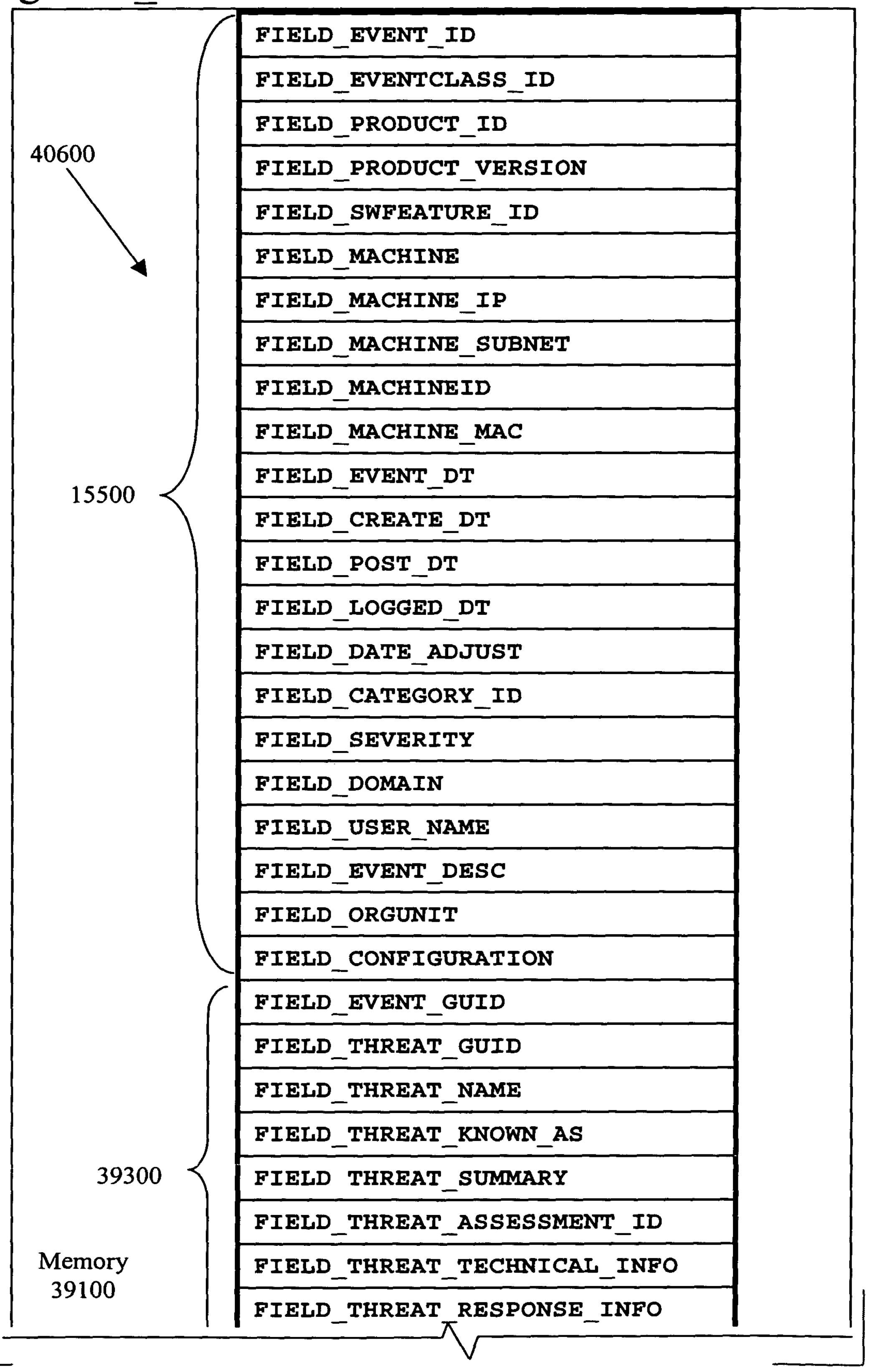
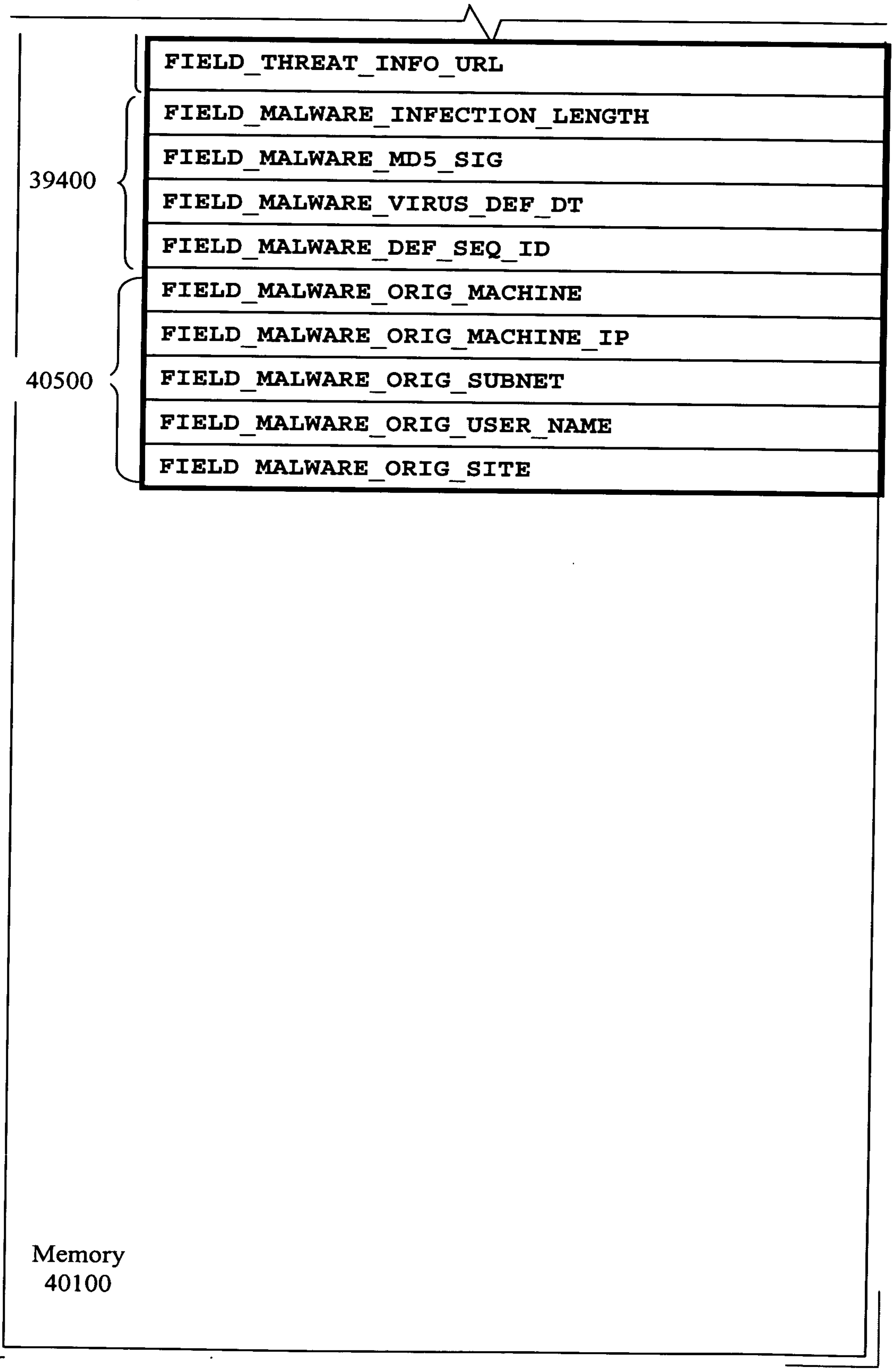

40600
15500
39300
Memory
39100
FIELD_EVENT_ID
FIELD_EVENTCLASS_ID
FIELD_PRODUCT_ID
FIELD_PRODUCT_VERSION
FIELD_SWFEATURE_ID
FIELD_MACHINE
FIELD_MACHINE_IP
FIELD_MACHINE_SUBNET
FIELD_MACHINEID
FIELD_MACHINE_MAC
FIELD_EVENT_DT
FIELD_CREATE_DT
FIELD_POST_DT
FIELD_LOGGED_DT
FIELD_DATE_ADJUST
FIELD_CATEGORY_ID
FIELD_SEVERITY
FIELD_DOMAIN
FIELD_USER_NAME
FIELD_EVENT_DESC
FIELD_ORGUNIT
FIELD_CONFIGURATION
FIELD_EVENT_GUID
FIELD_THREAT_GUID
FIELD_THREAT_NAME
FIELD_THREAT_KNOWN_AS
FIELD THREAT_SUMMARY
FIELD_THREAT_ASSESSMENT_ID
FIELD_THREAT_TECHNICAL_INFO
FIELD_THREAT_RESPONSE_INFO

Fig. 40B_2


FIELD_THREAT_INFO_URL
FIELD_MALWARE_INFECTION_LENGTH
FIELD_MALWARE_MD5_SIG
FIELD_MALWARE_VIRUS_DEF_DT
FIELD_MALWARE_DEF_SEQ_ID
FIELD_MALWARE_ORIG_MACHINE
FIELD_MALWARE_ORIG_MACHINE_IP
FIELD_MALWARE_ORIG_SUBNET
FIELD_MALWARE_ORIG_USER_NAME
FIELD MALWARE_ORIG_SITE
39400
40500
Memory
40100

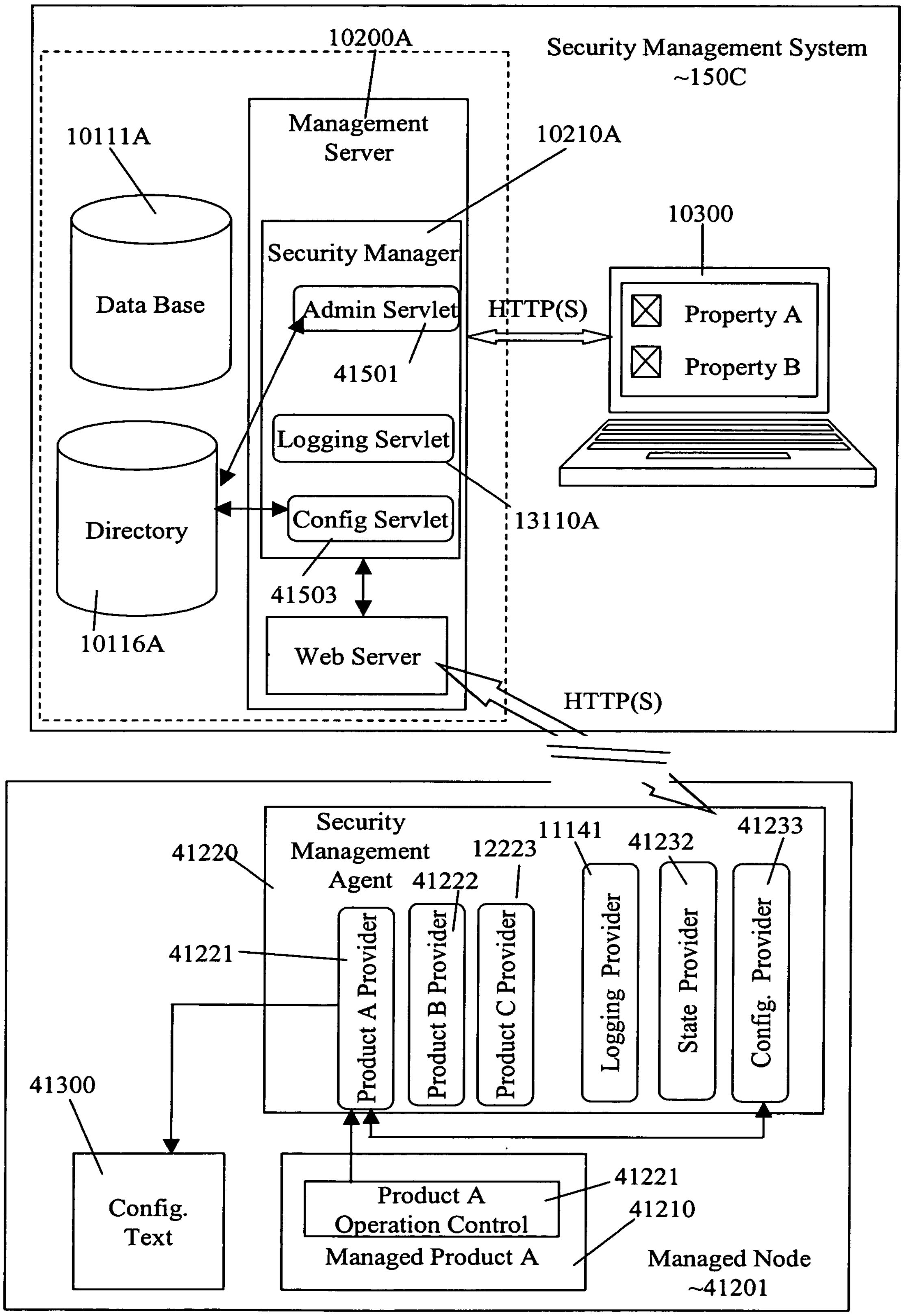

10200A
Security Management System
~150C
10111A
Management Server
10210A
Data Base
10300
Security Manager
Admin Servlet
HTTP(S)
Property A
Property B
41501
Logging Servlet
Directory
Config Servlet
13110A
41503
10116A
Web Server
HTTP(S)
41220
Security Management Agent
11141
12223
41232
41233
41222
41221
Product A Provider
Product B Provider
Product C Provider
Logging Provider
State Provider
Config. Provider
41300
Config. Text
Product A Operation Control
41221
41210
Managed Product A
Managed Node
~41201

Fig. 41

42000

```
        <?xml version="1.0" encoding="UTF-8" ?>
- <SesaIntegrationData xmlns:xsi=
        "http://www.w3.org/2001/XMLSchema-instance"
       xsi:noNamespaceSchemaLocation="NabooBase.xsd">
42001 -<SesaProductData>
          <Version>1.00</Version>
          <Author>ABC</Author>
          <Revision>0.01</Revision>
          <RevDate>Feb 15 2003</RevDate>
          <Product>
          <!-- Simple Sample Product  -->
          <!--  Product ID 3001  ==> Numbering
              Range [30,010,000 - 300,019,999]  -->
            -<Product Id="3001">
              <Version>4.0</Version>
              <Vendor>Symantec Corporation</Vendor>
              <SKUNumber>1234</SKUNumber>
              <Caption>Sample Product</Caption>
              <Description>Simple Sample Product
                     for NIK  </Description>
              <Name>Sample Product</Name>
              <DisplayName LangId="10001">
                            Sample</DisplayName>
              <EventFamilyMembership Id="90000" />
42002           <DataDefinition>
                <!--  Software Feature Ids
                           30,010,101-30,010,999 -->
                 - <SoftwareFeature Id="30010101">
                        <Caption>Sample Software
                             Feature </Caption>
                        <Description>Sample Software
                              Feature</Description>
42003                   <Name>30010101</Name>
                        <DisplayName LangId="10001">
                            Sample Software Feature
                            </DisplayName>
                        <FeatureRole>SESA_LOGGING
                            </FeatureRole>
                     </SoftwareFeature>
                </DataDefinition>
           </Product>
    </SesaProductData>
 </SesaIntegrationData>
```

Fig. 42

```
package com.symantec.management.example;

import java.io.*;
import java.util.*;
import java.net.*;

import org.snia.wbem.cim.CIMException;
import org.snia.wbem.cim.CIMNameSpace;
import org.snia.wbem.cim.CIMObjectPath;
import org.snia.wbem.cim.CIMValue;
import org.snia.wbem.client.CIMOMHandle;
import org.snia.wbem.client.CIMClient;

import com.symantec.management.providers.SESAProvider;
import com.symantec.management.providers.SymcObject;
import com.symantec.management.providers.SESAException;

/** example of the Agent/Provider extension interface
 */
public class ExampleProvider extends SESAProvider
{
        // product and feature IDs for the advanced sample
        public final int                ADV_SAMPLE_APP_PRODUCT_ID = 3002;
        public final int                ADV_SAMPLE_APP_FEATURE_ID = 30020101;

        // constants for the config properties
        public final String                     CFGPROP_POLLTIME = "PollTime";

        public final String                     PROVIDER_NAME =
"Symc_ExampleProvider";

// the local cache for your provider's configuration - name should be all lower case
        private final String     CONFIG_FILE_NAME = "exampleprovider.cfg";

        private Properties              m_props = null;// holds the config properties

        private CIMObjectPath                   m_cimPath = null;// object for the Example
Provider
        private CIMClient                       m_cimClient = null;// communicate with the
CIMOM

        public ExampleProvider()
        {
                System.out.println("--inside the ExampleProvider() constructor");
        }

        /*   perform initialization of the provider<p>
            Any initialization should be done in here. It is not necessary to request
```

Fig. 43A

```
         * configurations from the Config Provider, as those are retrieved automatically
         * and sent to applyConfig().
         */
        public void initialize(CIMOMHandle ch) throws CIMException
        {
                System.out.println("\n***inside the Example Provider!\n");

                // Any threads to be used should be created in here
                // Threads should be "daemon" so that the shutdown does not
                // have to wait on them too long.

                try
                {
                        loadConfigFile();
                }
                catch (CIMException e)
                {
                        System.out.println("Could not load example provider's configuration
file.");
                }

// There are other methods available in the Config Provider to get information
// about the machine and the Management Server. These are:
//      getMachineId() - the machine id from the bootstrap process
//      getDB() - get the DN, as used in the Directory
//      getDomain() - get the Domain that this machine was bootstrapped into
//      getOrgUnit() - get the OrgUnit the machine was bootstrapped into
//      getManagementServer() - returns a String for the URL to the Management Server
//      getManagementServerAddress() - returns the IP address of the Management Server
//      getManagementServerPort() - returns the port the Management Server listens on
//      getSecManagementServer() - returns a String for the URL to the Secondary
Management Server
//      getSecManagementServerAddress() - returns the IP address of the Secondary
Management Server
//      getSecManagementServerPort() - returns the port the Secondary Management
Server listens on
//      getUseSSL() - returns true or false, whether or not to use SSL

// There are also two more methods to retrieve configuration:
//      getConfig(int ProductId, int FeatureId)
//      getConfig(int ProductId, int FeatureIds[])
//      These methods both retrieve a String which contains the entire set of properties from
//      the Directory. The method illustrated below retrieves the same information, but the
data
//      is parsed for you and returned in a HashMap that looks something like this:

// see the ConfigParser class for a description of what the HashMap returned from
// getConfigProperties looks like
```

Fig. 43B

```
	}


	/** shut down the provider
	  */
	public void cleanup() throws CIMException
	{
		// destroy any threads created in initialize()
	}


	/** get the Service object that relates to this Provider
	 * @return Service
	 */
	public SymcObject getService()
	{
		try
		{
			if (m_Service == null)
			{
				// the file name referenced "ExampleProvider.svc" can be in
mixed case
				// as the constructor translates all file names to lower case.
				// The physical file on disk must be in lower case (enforced by
the Makefile)

				m_Service = new SymcObject("Symc_Service",
"ExampleProvider.svc");
			}
		}
		catch (SESAException se)
		{
			se.printStackTrace();
		}
		return (m_Service);
	}


	/** the CIM invokeMethod call
	  */
	public CIMValue invokeMethod(CIMObjectPath op, String name, Vector in, Vector
out)
		throws CIMException
	{
		System.out.println("ExampleProvider.invokeMethod('" + name + "')");

		CIMValue ret = new CIMValue("unrecognized method '" + name + "'");
```

Fig. 43C

```
        if (name.equalsIgnoreCase("getconfigdata"))
                ret = getConfigData(in, out);
        else if (name.equalsIgnoreCase("getconfigproperty"))
                ret = getConfigProperty(in, out);

        return (ret);
}


/** obtains all config settings for the application
 */
public CIMValue getConfigData(Vector in, Vector out)
{
        if (m_props == null)
                return (new CIMValue(""));

        String sRet = "";
        Enumeration enum = m_props.keys();
        while (enum.hasMoreElements())
        {
                String sKey = (String) enum.nextElement();
                String sVal = (String) m_props.get(sKey);
                if (sVal != null)
                        sRet += sKey + "=" + sVal + "\n";
        }

        if (sRet.length() == 0)
                sRet = "key=value\n";

        System.out.println("\n*****Returning config data: '" + sRet + "'");
        return (new CIMValue(sRet));
}


/** obtains a specific, named config property
 */
public CIMValue getConfigProperty(Vector in, Vector out)
{
        if (m_props == null)
                return (new CIMValue(""));

        String sParam = null;
        try
        {
                sParam = getParameterString(in, "propName");
        }
        catch (SESAException se)
```

Fig. 43D

```
                {
                        return (new CIMValue(""));
                }

                if (sParam == null)
                        return (new CIMValue(""));

                // Looks in the private cache of properties to retrieve the value.
                String sVal = (String) m_props.get(sParam);
                if (sVal == null)
                        sVal = "";

                if (sVal.length() == 0)
                        sVal = sParam + ".value";

                System.out.println("\n***Returning config property '" + sParam + "' = '" +
sVal + "'");
                return (new CIMValue(sVal));
        }


        /** Used for sending messages between Providers.<P>
         * This method is called from another Provider to inform this Provider of a specific
         * event. The string contains information that can be parsed.
         * @param msg The String representing the message.
         */
        public void sendMessage(String msg)
        {
        }

        /** get the name of the Provider
         * @return a string representing the name of this Provider
         */
        public String getName()
        {
                return (PROVIDER_NAME);
        }


        /** Informs the Provider of an updated configuration.<P>
         * This is called from the Configuration Provider when there is a new
         * configuration that the Provider should use.
         * @param newConfigs A HashMap representing the configuration properties that
         *              the Provider should use from this point forward. See the ConfigParser
         *              class for a description of the contents of the HashMap
         */
        public void applyConfig(HashMap newConfigs)
        {
```

Fig. 43E

```
                // Get the properties that we are interested in from the entire set.  The entire
set includes
                // all products and features.  We are interested only in our own application.
                Properties newProps = getCfgPropertySet(newConfigs,
ADV_SAMPLE_APP_FEATURE_ID, "SampleApplication");
                if (newProps == null)
                {
                        System.out.println("no properties found for SWF " +
getSoftwareFeatureId());
                        return;
                }
                // Private cache of properties.
                m_props = newProps;

                String sVal = (String) m_props.get(CFGPROP_POLLTIME);
                if (sVal != null)
                {
                        try
                        {
                // Set the poll time in the instance so that the application can use the standard
CIM
                // call getProperty to retrieve it.  This also allows the poll time to become
part of
                // the application's set of state variables. You would put a property here only
if you
                // want it accessible through CIM.  If not, keep it only in the private cache.
                                createCIMClient();
                                m_cimClient.setProperty( m_cimPath, "ProviderPollTime",
new CIMValue(sVal) );
                        }
                        catch (CIMException ce)
                        {
                                System.out.println("\n>>>>> Error setting property
ProviderPollTime in instance\n" );
                                System.out.println( ce.toString() );
                        }
                }

                // update the local configuration file from the HashMap
                File fiConfig = new File(CONFIG_FILE_NAME);

                FileOutputStream os = null;

                try
                {
                        os = new FileOutputStream(fiConfig);
                        m_props.store(os, null);
                }
```

Fig. 43F

```
                catch (FileNotFoundException fnf)
                {
                }
                catch (IOException ioe)
                {
                }

                if (os != null)
                        try
                        {
                                os.close();
                        }
                        catch (IOException ioe)
                        {
                        }

// At this point the provider could communicate with its application to push the new
configuration
// to the application.  Alternatively, the application could poll for changes to the
configuration
// file created above and read its configuration from the file.

// Check to see if the advanced sample application is listening on the pre-defined port
                try
                {
                        // The port is defined in the advanced sample app.
                        Socket sock = new Socket("127.0.0.1", 4990);

                        if (sock != null)
                        {
                        // If we're able to get a connection, then the advanced sample app is
running
                        // and waiting for a connection on localhost:4990.  Attempt to write
out the data.
                        // The data will be the value that was assigned to "ProviderPollTime".
                        // Note that the data sent can also be custom xml settings that will get
parsed
                        // by the application.
                                BufferedOutputStream ostream = new
BufferedOutputStream(sock.getOutputStream());

                                ostream.write(sVal.getBytes());

                                // Cleanup
                                ostream.close();
                                sock.close();
                        }
                }
```

Fig. 43G

```
                catch (UnknownHostException e)
                {
                }
                catch (IOException e)
                {
                }
                catch (Exception e)
                {
                }
        }


        /** get the Product ID for this Provider
         * @return an integer representing the Product ID that this Provider is associated
with.
         */
        public int getProductId()
        {
                return (ADV_SAMPLE_APP_PRODUCT_ID);
        }


        /** get the Software Feature ID
         * @return an integer representing the Software Feature ID that this Provider is
         *                      associtated with, or 0 if the Provider is not associated with a
specific
         *                      feature ID.
         */
        public int getSoftwareFeatureId()
        {
                return (ADV_SAMPLE_APP_FEATURE_ID);
        }


        /** load the configuration file from disk
          */
        private void loadConfigFile() throws CIMException
        {
                File fiConfig = new File(CONFIG_FILE_NAME);
                if (!fiConfig.exists())
                        throw new CIMException("Symc_ExampleProvider: cannot find
config file '" + CONFIG_FILE_NAME + "'");

                m_props = new Properties();
                try
                {
                        m_props.load(new FileInputStream(fiConfig));
                }
```

Fig. 43H

```
		catch (FileNotFoundException fnf)
		{
			throw new CIMException("Symc_ExampleProvider: cannot load
config file '" + CONFIG_FILE_NAME + "'");
		}
		catch (IOException ioe)
		{
			throw new CIMException("Symc_ExampleProvider: error reading
configuration file");
		}
	}

	private void createCIMClient() throws CIMException
	{
		if (m_cimPath == null)
		{
			m_cimPath = new CIMObjectPath( PROVIDER_NAME );
			m_cimPath.addKey( "CreationClassName", new CIMValue(
"Symc_Service" ) );
			m_cimPath.addKey( "Name", new CIMValue(
"30020101.exampleprovider" ) );
			m_cimPath.addKey( "SystemCreationClassName", new CIMValue(
"Symc_ComputerSystem" ) );
			m_cimPath.addKey( "SystemName", new CIMValue( "localhost" ) );
		}

		if (m_cimClient == null)
		{
			CIMNameSpace cns = new CIMNameSpace( "localhost", "root" );
			m_cimClient = new CIMClient( cns, null, null, CIMClient.LOCAL );
		}
	}
}
```

Fig. 43I